

NuiMod - Um ambiente para criação de interfaces usando poses e gestos

Jordano Ribeiro Celestrini¹, Crediné Silva de Menezes¹

¹Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, s/n, Vitória – ES – Brazil
jordanorc@gmail.com, credine@inf.ufes.br

Abstract. *The usage of motion sensors to educational applications is a growing practice, boosted by the emergence of new devices like Kinect. This type of interface provides a playful way for student interaction with objects of knowledge, making the learning process interesting and enjoyable. However, in the solutions development context, support for modeling using natural gestures and interfaces in a web environment is still a gap to be filled. This article presents the NuiMod, an environment for modeling poses and gestures whose intent is to facilitate the building and use of gesture models, as well as providing the use of Natural User Interfaces in a web environment.*

Resumo. *A utilização de sensores de movimento para aplicações educacionais é uma prática crescente, potencializada pelo surgimento de dispositivos de movimento como o Kinect. Este tipo de interface fornece um meio lúdico para a interação dos estudantes com objetos de conhecimento, tornando o processo de aprendizagem interessante e prazeroso. Contudo, no contexto de desenvolvimento de soluções, o suporte à modelagem de gestos e utilização de interfaces naturais em ambiente web ainda é uma lacuna a ser preenchida. Este artigo apresenta o NuiMod, um ambiente para modelagem de poses e gestos cujo objetivo é facilitar a criação e utilização de modelos gestuais, bem como proporcionar o uso de Interfaces Naturais de Usuário em ambiente web.*

1. Introdução

Com a evolução dos computadores, surgiram também novas formas de interação humano-computador. O surgimento de dispositivos de baixo custo fez emergir uma nova forma de interação - as interfaces naturais de usuários (NUI) - caracterizadas pela inexistência no contato com o dispositivo de interação, permitindo ao usuário controlar equipamentos por meio de gestos corporais.

No contexto dos jogos digitais, a evolução das interfaces naturais tem sido percebida de maneira mais acentuada, com o lançamento de novos sensores que melhoram a interação homem-computador. Após o lançamento do Kinect, surgiram as primeiras iniciativas para utilização deste tipo de sensor, extrapolando o universo dos consoles de jogos digitais como, por exemplo, para navegação em mapas, proposto por [Kamel Boulos et al. 2011] e no apoio ao ensino, como proposto em [Silva e Neto 2012], [Lee, Liu e Zhang 2012] e [Villaroman, Rowe e Swan 2011].

Para facilitar o desenvolvimento de aplicações computacionais utilizando NUI, nasceram as primeiras iniciativas implementadas como *frameworks*. A despeito do seu pioneirismo na criação de interfaces NUI, tais *frameworks* apresentam deficiências como dificuldades quanto a sua utilização em conjunto com movimentos complexos, embora existam abordagens que tenham demonstrado seu bom funcionamento em cenários controlados. Além disso, um outro ponto negativo é inexistência de ambientes voltados especificamente para a modelagem de poses e gestos, que é uma tarefa essencial em contextos como o da fisioterapia.

Na educação, o desenvolvimento de aplicações é fortemente dependente de

interfaces e da facilidade com se pode construí-las. Neste sentido, o surgimento de novas ferramentas que reduzam o esforço do usuário no processo de concepção e adaptação de interfaces apresentam-se como uma oportunidade para sua melhor utilização. Esta iniciativa já pode ser percebida no contexto de jogos digitais, com a utilização das NUI na tentativa de aproximar os jogadores das interfaces construídas. Na área educacional, busca-se corroborar com essa abordagem a partir da criação de novos ambientes que reduzam o esforço para a construção de interfaces naturais de usuário.

O objetivo deste trabalho é apresentar o NuiMod, um ambiente para criação de interfaces utilizando gestos e poses. Sua arquitetura está dividida em 3 componentes principais: o NuiMod Connector, responsável pela comunicação com os dispositivos de interação natural, o NuiMod Engine, responsável pelo processamento dos dados e o NuiMod Editor, responsável pela edição de gestos e poses. O roteiro metodológico seguido na elaboração deste estudo foi: revisão bibliográfica; projeto e prototipação do ambiente; proposição de cenários de aplicação e teste.

Este trabalho está dividido da seguinte maneira: na próxima seção, inicia-se a discussão sobre o problema. Segue-se então para a descrição de trabalhos correlatos. A seção 4 aborda o NuiMod e os detalhes de sua arquitetura. A seção seguinte trata sobre os principais cenários para aplicação do NuiMod e, então, a contribuição do trabalho e suas limitações.

2. Facilitando a Interação

Os recentes avanços na interação homem-computador têm facilitado o desenvolvimento de interfaces naturais de usuário, proporcionando maneiras mais intuitivas de interagir com os dispositivos [Villaroman, Rowe e Swan 2011]. As pesquisas na área de Interfaces Naturais de Usuário são relativamente novas e procuram diminuir a curva de aprendizagem entre o usuário e as interfaces construídas, traduzindo movimentos corporais em ações. Segundo [Blake 2011], uma interface natural é desenvolvida para reutilizar habilidades existentes e interagir de maneira apropriada com o conteúdo.

No contexto educacional a utilização de NUI pode proporcionar ao aluno um meio direto para interagir com seus objetos de aprendizagem, uma vez que permite sua exploração de maneira natural, com interações a partir de gestos e poses.

Uma das áreas na qual este tipo de prática pedagógica pode ser utilizada é no ensino da anatomia humana que, segundo [Fornaziero et al. 2010], precisa ser repensada a fim de corresponder às expectativas deste novo e atual momento de constante evolução da tecnologia educacional.

As NUI apoiam esta iniciativa por meio de modelos anatômicos articulados, que ajudam a compreender os movimentos humanos e a relação de interdependência entre seus diversos sistemas constituintes. Tais modelos podem controlados por meio de gestos, seguindo o movimento de seu utilizador para, por exemplo, apoiar a análise da musculatura corporal ou do sistema respiratório e cardiovascular durante uma determinada atividade, como um salto ou agachamento.

Neste sentido, um problema que ainda precisa ser solucionado é a disponibilização desses modelos anatômicos em um ambiente de fácil acesso, como a Internet, e que permita aos usuários modelar seus próprios gestos e poses. Tal intento poderia contribuir com uma visão mais crítica e criativa para os estudantes, aproximando-os de seu objeto de estudo.

O processamento computacional de linguagem natural requer a solução de três questões não triviais: detecção, rastreamento e compreensão de comportamento [Suma et al. 2013]. Uma das dificuldades ao se desenvolver interfaces naturais de usuário é a modelagem de poses e gestos. Grande parte dos sistemas que utilizam este tipo de interface implementam gestos e poses pré-definidas como em [Silva e Neto 2012] e

[Lee, Liu e Zhang 2012], e não permitem sua personalização. Além disso, as soluções existentes são predominantemente limitadas ao ambiente desktop.

São exemplos de ambientes para suporte de construção de sistemas com uso de Interfaces Naturais de Usuário: Flexible Action and Articulated Skeleton Toolkit (FAAST), Xkin e o KinectJS. Todavia, nenhuma destas soluções possui enfoque na modelagem de gestos e poses de maneira intuitiva. A modelagem, nestes ambientes, requer conhecimento expressivo nas ferramentas utilizadas, o que inviabiliza o seu uso por grande parte dos usuários. Deste modo, a construção de um ambiente para modelagem de poses e gestos preenche a lacuna existente neste universo de estudo.

3. Trabalhos correlatos

Esta seção apresenta uma revisão bibliográfica sobre os principais trabalhos desenvolvidos com o objetivo de facilitar o uso de NUI em contextos que envolvam o uso de computadores. Após a apresentação dos trabalhos são evidenciados os pontos positivos e negativos de cada uma das soluções.

3.1. FAAST

O FAAST (Flexible Action and Articulated Skeleton Toolkit) é um *framework* criado para facilitar a integração entre dispositivos de interação natural e aplicações de realidade virtual e jogos utilizando sensores de profundidade em conformidade com o padrão OpenNI [Suma et al. 2011]. Por meio de uma câmera com sensor de profundidade, utilizando luz infravermelha, ele manipula duas grandes categorias de informação: ações e esqueletos articulados.

A arquitetura do FAAST, observada na figura 1, consiste em módulos que lêem dados de esqueleto, um módulo de reconhecimento de ação, um módulo emulador que envia eventos simulados de teclado e mouse para a janela ativa, e um servidor para transmitir a posição e orientação de cada articulação esqueleto para aplicações de rede [Suma et al. 2013].

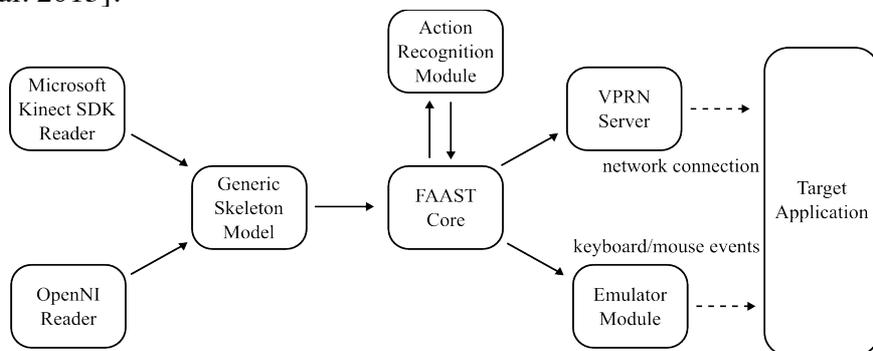


Figura 1. Arquitetura FAAST. Adaptado de [Suma et al. 2013]

O FAAST suporta o sensor Kinect usado no Microsoft Kinect for Windows SDK e os sensores que façam uso da biblioteca de rastreamento de esqueleto NITE da PrimeSense [Villaroman, Rowe e Swan 2011]. A comunicação com cada rastreador do esqueleto é dividido em módulos separados que são selecionados e carregados em tempo de execução, fazendo com que cada módulo leia os dados a partir de seu respectivo rastreador numa estrutura de dados de esqueleto genérico [Suma et al. 2013].

3.2. XKin

O Xkin é um *framework* para Kinect que permite uma comunicação entre humanos e computadores por meio das mãos de maneira natural e intuitiva. O projeto possui código aberto, licenciado em FreeBSD e implementado na linguagem C [Pedersoli et al. 2012]. Ele foi desenvolvido sobre a biblioteca libfreenect e faz uso do OpenCV para lidar com matrizes digitais, imagens e vídeos. Sua composição se baseia em quatro

bibliotecas principais: libbody, libhand, libposture and libgesture, conforme pode ser visto na figura 2.

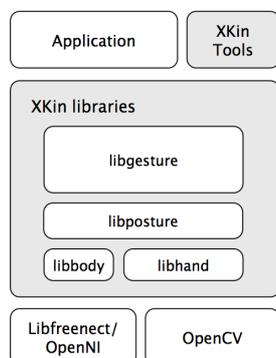


Fig. 2 Arquitetura do XKin [Pedersoli et al. 2012]

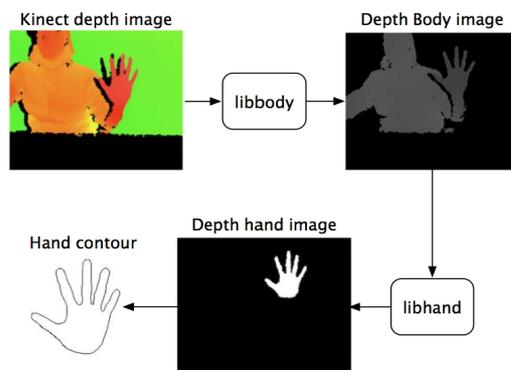


Fig.3 - Detecção do contorno das mãos com XKin [Pedersoli et al. 2012]

A **libbody** é utilizada para detectar o corpo de uma pessoa na cena profundidade adquirida pelo dispositivo (Kinect). Por meio dela, pode-se processar a imagem para isolar a área correspondente ao corpo e definir para zero todos os pixels que pertencem ao fundo.

Para detecção e extração do contorno das mãos, utiliza-se a **libhand**. Este processo é feito em três etapas: por meio da função *hand_detection*, obtém-se uma máscara binária que representa a forma da mão e sua imagem de profundidade. Os dados brutos do contorno da mão são então extraídos por meio da função *get_hand_contour_basic*. Estes dados são processados pela função *get_hand_contour_advanced*, que utiliza uma abordagem mais sofisticada para conseguir um contorno mais definido da mão (figura 3).

A **libposture** provê funções para classificar poses estáticas das mãos. O XKin fornece duas abordagens de classificação. A abordagem simples é extremamente rápida e robusta, mas limita-se ao reconhecimento de apenas duas poses: abrir e fechar as mãos. A classificação avançada distingue conjunto maior de posturas, porém é menos robusta se comparado à técnica anterior.

A classificação da trajetória do movimento da mão é realizado por meio da **libgesture**, implementando uma estrutura flexível para lidar com Modelos Ocultos de Markov (HMM) e os algoritmos típicos para os modelos de treinamento e testes.

3.3. KinectJS

O projeto KinectJS tenta combinar JavaScript e Kinect para interagir com um site, com o objetivo final de trazer controles de movimento em HTML5 [Noyes 2006], resultando numa biblioteca composta por dois módulos. O **Kinect.js** é um módulo cliente que possui código aberto, desenvolvido em Javascript e que possibilita operar o Kinect por meio de uma API [Kalogiros 2012]. E o **KinectSocketServer**, módulo servidor desenvolvido com licença proprietária que possui acesso direto ao hardware por meio do Kinect SDK. A comunicação entre eles ocorre via websockets.

É possível realizar o rastreamento de movimentos para até duas pessoas ao mesmo tempo, bem como controlar o motor do Kinect pelo navegador. O *framework* possui gestos pré-definidos, como furto, saltar e fugir. Todavia, os gestos são simples e muitas vezes exigem grandes movimentos. Infelizmente, as interfaces com o usuário em geral não estão adaptadas para interação natural, mas é, de fato fortemente inspirado por novas interfaces, como o Metro UI da Microsoft[12].

3.4. Conclusões

As tecnologias discutidas para o desenvolvimento de aplicações com interfaces naturais de usuário satisfazem os requisitos de detecção, rastreamento e compreensão de comportamento humano. No entanto, todas elas são voltadas para ambientes desktops. Sinteticamente, as características podem ser reunidas da seguinte maneira:

Tabela 1 - Síntese de funcionalidades dos frameworks

	Características positivas	Características negativas
FAAST	<ul style="list-style-type: none"> - Simples utilização - Arquitetura flexível 	<ul style="list-style-type: none"> - Código proprietário - Não possui ferramenta para modelagem de gestos - Não possui suporte à uso na web
Xkin	<ul style="list-style-type: none"> - Código aberto - Comunicação via websocket 	<ul style="list-style-type: none"> - Não possui ferramenta para modelagem de gestos - Não possui suporte à uso na web - Suporte à rastreamento apenas das mãos
KinectJS	<ul style="list-style-type: none"> - Pode ser utilizado na web 	<ul style="list-style-type: none"> - API (KinectSocketServer) de acesso ao hardware em código proprietário - Não possui ferramenta para modelagem de gestos

4. Solução proposta: NuiMod

Tendo em vista a limitação das tecnologias atuais voltadas para o desenvolvimento de sistemas de linguagem natural, no que diz respeito à modelagem gestual, foi concebido o NuiMod, levando em consideração os seguintes requisitos: ambiente web; extensível a novos dispositivos; multiplataforma; capaz de modelar gestos. Ele é uma proposta de ambiente de modelagem gestual, composto por três componentes: o NuiMod Editor, o NuiMod Connector e o NuiMod Engine. A figura 4 representa o ambiente e suas dependências.

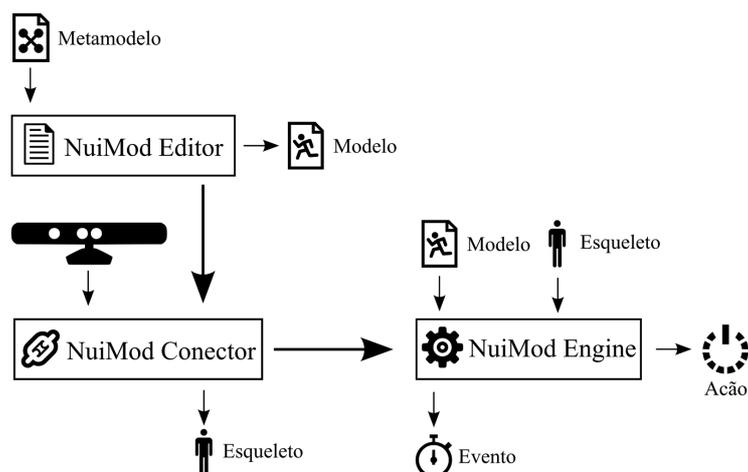


Figure 4. Arquitetura NuiMod

4.1. NuiMod Connector

O NuiMod Connector é o componente responsável por transmitir os dados do Kinect para a aplicação web. Implementado em linguagem C#, comunica-se com o Kinect por

meio do *Kinect for Windows SDK*. Faz uso de um protocolo de comunicação baseado em troca de mensagens, realizada via websockets. O protocolo websocket permite a comunicação bidirecional entre um cliente executando um código não confiável, em um ambiente não controlado, e um host remoto [Fette e Melnikov 2011].

Os comandos a serem enviados ao NuiMod Connector podem ser de dois tipos: síncronos ou assíncronos. Os comandos síncronos incluem aqueles cuja resposta é enviada ao cliente assim que executado. Nesta categoria estão os comandos de listagem e inicialização de dispositivos, e término de rastreamento do esqueleto.

O comando de início de rastreamento do esqueleto é um assíncrono, uma vez que irá enviar respostas ao cliente até que o comando de término de rastreamento do esqueleto seja recebido pelo servidor. A figura 5, ilustra os comandos e respostas emitidas pelo NuiMod Connector.

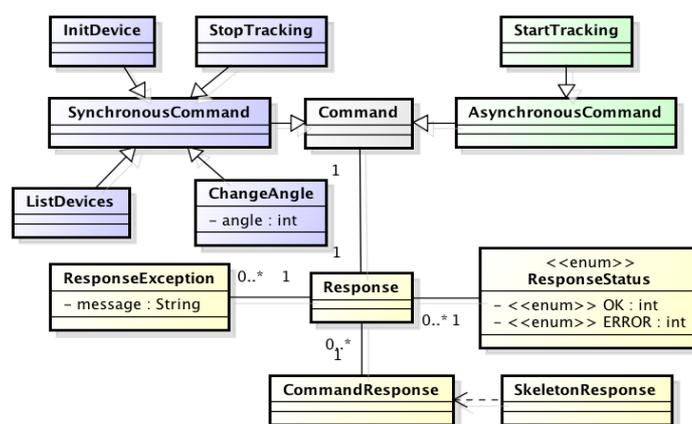


Figura 5. Tipos de comandos e respostas do módulo NuiMod Connector

As informações são transmitidas entre o cliente e o servidor por meio de mensagens JSON, um formato leve para intercâmbio de dados computacionais. O cliente deve informar, ao enviar uma mensagem, qual o tipo de comando e, quando houver, quais os parâmetros para a execução daquela instrução. Adiante (fig. 6) um exemplo de mensagem para o comando *ChangeAngle*, que altera a inclinação do Kinect:

```
{
  "ChangeAngle": {
    "angle": "20"
  }
}
```

Figura 6. Exemplo de mensagem para envio de comando ao NuiMod Connector

No cenário proposto, o Connector pode ser instalado na máquina cliente a partir de um arquivo executável ou inicializado diretamente pelo navegador, por meio do ClickOnce Deployment, tecnologia de implantação que permite a criação de aplicativos AutoAtualizáveis, que podem ser instalados e executados com mínima interação do usuário, por meio de um link na web.

4.2. NuiMod Engine

O NuiMod Engine é o componente responsável pelo processamento dos dados recebidos do NuiMod Connector. Para suportar esta tarefa, os dados recebidos a partir do sensor são armazenados em um histórico, que é utilizado na comparação de similaridade entre as poses. Uma vez que o Kinect é capaz de transmitir trinta quadros por segundo, cada novo dado sobre o esqueleto recebido pela engine é armazenada em uma variável, e os dados antigos são eliminados. Por padrão, a engine suporta o armazenamento de 10 segundos de movimentação, que equivalem a 300 frames. Isto é

suficiente para o processamento da maior parte dos gestos, no entanto este valor pode ser personalizado de acordo com a necessidade do utilizador. Como entrada, a Engine recebe um arquivo .mdd que é exportado pelo NuiMod Editor.

Para o reconhecimento de poses, a cada novo frame recebido, o sistema calcula a similaridade entre as posições das articulações da pose desejada e a posição das articulações no frame. Em termos matemáticos, simetria consiste em um espaço métrico (X, d) , onde X é um conjunto munido de uma métrica, ou seja, uma função $d: X \times X \rightarrow \mathbb{R}$ tal que, para quaisquer $x, y, z \in \mathbb{R}$:

- $d(x, y)$ é um número real, não negativo
- $d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) = d(y, x) = 0$ (simetria)
- $d(x, z) \leq d(x, y) + d(y, z)$

É possível determinar as poses similares e diferentes a partir da definição da função distância. Aquelas que possuem distâncias pequenas são consideradas similares. A escolha da função d encerra um grau de liberdade muito grande, o que dificulta sua escolha [De Oliveira 2006]. O conjunto de articulações para o qual foi definida a função d é determinado como espaço métrico, representado pelo par $\{\text{conjunto}, \text{distância}\}$.

Assim, para efetuar o reconhecimento de similaridade é necessário especificar um espaço métrico, em símbolos: $\langle J, d \rangle$; onde J é o conjunto de articulações consideradas e d a função distância do espaço métrico conhecido como função de similaridade.

A abordagem mais simples para esta finalidade é usar uma métrica de Minkowski, tais como a Distância Euclidiana [Parvini e Shahabi 2007]. A Distância Euclidiana entre dois vetores (neste caso, pode-se interpretar cada vetor como a posição de uma articulação do esqueleto) $|a_1, a_2, a_3, a_n|T$ e $|e_1, e_2, e_3, e_n|T$ é definida por:

$$\sqrt{\sum_{i=1}^n (a_i - e_i)^2}$$

Como os dados de articulação são tridimensionais (horizontal, vertical e de profundidade), a distância euclidiana é computada como:

$$\sqrt{(a_x - e_x)^2 + (a_y - e_y)^2 + (a_z - e_z)^2}$$

onde a é a articulação base e e é a articulação atual. Logo, para cada frame recebido calcula-se a distância euclidiana para cada uma das 20 articulações rastreadas. A Engine considera similares aquelas articulações cuja distância euclidiana esteja dentro do desvio máximo informado pelo utilizador. O utilizador pode ainda configurá-la para ignorar os cálculos seguintes assim que encontrar uma articulação cuja distância euclidiana esteja além dos limites estabelecidos no Editor.

O reconhecimento de gestos é realizado de maneira similar. Como um gesto é composto por várias poses, a Engine processa cada frame recebido pelo Connector comparando-a com a pose atual do gesto desejado. Caso sejam similares, a Engine automaticamente passa à pose seguinte até que todas as poses do gesto tenham sido executadas. Se o gesto exceder o tempo limite para ser executado indicado pelo modelador, a Engine inicia novamente o processamento a partir da pose inicial.

4.3. NuiMod Editor

Desenvolvido em Python e Django, o NuiMod Editor é um ambiente web para modelagem de poses e gestos. Python é uma linguagem de programação orientada a objetos, interpretada, fortemente tipada e dinâmica, focada em legibilidade de código com uma sintaxe clara e concisa [Lutz 2013]. O *framework* Django é importante implementação em Python, bastante utilizado para desenvolvimento rápido de aplicações web [Santana e Galesi 2010]. Ele segue o padrão de desenvolvimento Model-Template-View (MTV), que possui algumas semelhanças com o padrão MVC - Model-View-Controller, fornecendo separação clara de tarefas e responsabilidades entre os aspectos importantes de um aplicativo [Alchin, Kaplan-Moss e Vilches 2009].

Dois conceitos fundamentais para o uso da ferramenta são os projetos e os componentes. No NuiMod Editor, o usuário pode criar um projeto e adicionar componentes a ele; atualmente é possível utilizar dois tipos de componentes: gestos e poses.

O Editor possui 2 modos de funcionamento: manual e guiado. No modo manual, o usuário manipula um avatar articulado para modelar seus componentes (poses ou gestos), conforme pode ser visto na figura 7. No método guiado, o usuário pode utilizar o próprio sensor de movimentos para manipular o avatar. Para cada componente criado deve-se vincular um evento, que será disparado pela Engine assim que for realizado uma pose ou gesto.

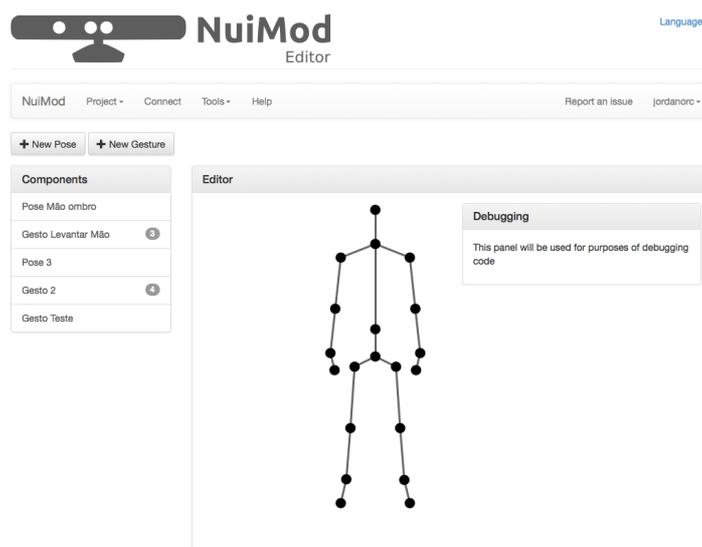


Figure 7. Tela de edição do NuiMod Editor

O usuário pode ainda estipular quais articulações deverão ser rastreadas em determinado componente, uma vez que usualmente um componente não utiliza todas as articulações do corpo. Além disso, rastrear articulações desnecessárias a determinado componente pode diminuir a taxa de acerto da Engine.

Ao finalizar o projeto, o usuário pode exportá-lo por meio do menu *Ferramentas* -> *Salvar no computador*. O sistema gera um arquivo XML, com a extensão *.nmd*, contendo os componentes criados pelo modelador e os eventos que devem ser disparados assim que o usuário realizar a pose ou gesto modelado.

O Editor disponibiliza ainda uma API para acesso aos dados por meio de Transferência de Estado Representativo (REST). Isto possibilita que aplicações terceiras consumam os componentes modelados pelo editor sem a necessidade de exportação do arquivo *nmd*.

5. Cenários para aplicação

Nesta seção serão apresentadas publicações que utilizam interfaces naturais de usuário na área educacional para apoio ao ensino. Existe um amplo acervo de trabalhos na área médica como [Gallo, Placitelli e Ciampi 2011], [González-Ortega et al. 2014] e [Ganesan e Anthony 2012], no qual o NuiMod também pode ser aplicado. Todos estes projetos foram criados para utilização em ambiente desktop e não permitem, de maneira geral, que os usuários modelem seus próprios gestos. O NuiMod aplicado a tais contextos poderia ampliar e facilitar o acesso à essas ferramentas, tornando mais lúdico o aprendizado.

Uma interface natural de interação para mapas é apresentada no trabalho de [Kamel Boulos et al. 2011]. O Kinoogle, nome pelo qual o software foi batizado, possibilita a navegação pelo Google Earth por meio do Kinect, utilizando OpenNI para coletar e processar informações de movimentação do usuário. O sistema faz uso do FAAST para mapear os movimentos do usuário em ações dentro do Google Earth. Neste mesmo contexto, pode-se aplicar o NuiMod ao Google Maps, a fim de possibilitar a navegação em mapas pela Internet.

Uma ferramenta de apoio aos docentes em sala de aula é idealizada em [Silva e Neto 2012], por meio do desenvolvimento de um protótipo de software utilizando Kinect. O projeto, batizado de Just Moving foi projetado a partir de um levantamento de gestos e funcionalidades sugeridos por professores da rede privada, pública e federal.

Para o ensino da matemática, é proposto o XDigit [Lee, Liu e Zhang 2012], um jogo que utiliza interface natural para interação com o usuário. O jogo - divertido e interativo - possui a temática do espaço, onde o jogador tem por objetivo combinar números através de operações aritméticas para encontrar o número de destino antes que o tempo se esgote. Segundo os autores, o trabalho foi motivado pela dificuldade na aprendizagem da disciplina, frequentemente relatada e conhecida entre as crianças. A maneira de ensinar matemática praticamente não mudou ao longo do tempo e, com o surgimento de novas tecnologias como tablets, smartphones e sensores de movimento, as pessoas têm demonstrado enorme interesse e motivação para desenvolver novos métodos interativos para o ensino.

[Morais et al. 2012], explora o uso dos sensores de movimento com aplicações para a educação a distância (EaD), em salas de aulas tradicionais, no ensino básico.

6. Considerações finais

O uso de sensores de movimento, como o Kinect, em diferentes áreas do conhecimento corresponde a um campo de pesquisa crescente e consolida estes dispositivos como uma interface para uso em finalidades que extrapolem os jogos digitais. Porém, a falta de uma plataforma de desenvolvimento integrada para modelagem e prototipação desse tipo de aplicação ainda representa uma barreira na sua utilização.

Apesar de algumas ferramentas facilitarem o desenvolvimento, ainda há um rol expressivo de limitações que precisam ser superadas, principalmente aquelas relacionadas à complexidade de utilização e a inexistência de plataformas para desenvolvimento de aplicações além do contexto de jogos digitais.

O desenvolvimento de aplicativos por usuários não especialistas em programação torna-se possível por meio de plataformas que reúnam características, metodologias e ferramentas de apoio para simplificar esta atividade. Os esforços nesta área ainda são incipientes constituindo-se, portanto, um campo promissor para pesquisa e desenvolvimento de interfaces baseadas em sensores de movimento e, particularmente usando o equipamento Kinect.

Neste sentido, ambientes como o NuiMod colaboram para a propagação de interfaces naturais de usuário, facilitando a modelagem de poses e gestos e contribuindo

para a incorporação desta tecnologia em ambientes web. Apesar dos resultados alcançados, algumas limitações a serem superadas podem ser observadas, como a criação de ferramentas de código-aberto e multiplataforma para rastreamento de esqueleto, além da ampliação dessa arquitetura para dispositivos móveis. Na área educacional, o próximo iniciativa é a integração do NuiMod em ambientes virtuais de aprendizagem, como o Moodle, bem como sua utilização para a construção de jogos educacionais para a web, reimplementando soluções como o Xdigit.

Referências bibliográficas

- ALCHIN, M.; KAPLAN-MOSS, J.; VILCHES, G. Pro Django. [S.l.]: Springer, 2009.
- BLAKE, J. The natural user interface revolution. *Natural User Interfaces in .NET*, Manning, p. 1–35, 2011.
- DE OLIVEIRA MARIN, L. Métodos Estatísticos no Reconhecimento de Faces. *Revista Eletrônica de Sistemas de Informação* ISSN 1677-3071 doi: 10.5329/RESI, v. 5, n. 2, 2006.
- FETTE, I.; MELNIKOV, A. The websocket protocol. 2011. Disponível em: <<http://tools.ietf.org/html/rfc6455>>. Acessado em 10/06/2014
- FORNAZIERO, C. C. et al. O ensino da anatomia: integração do corpo humano e meio ambiente. *Rev Bras Educ Med*, v. 34, n. 2, p. 290–297, 2010.
- GALLO, L.; PLACITELLI, A. P.; CIAMPI, M. Controller-free exploration of medical image data: Experiencing the Kinect. 2011, [S.l.: s.n.], 2011. p. 1–6.
- GANESAN, S.; ANTHONY, L. Using the kinect to encourage older adults to exercise: a prototype. *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*. Austin, Texas, USA: ACM. , 2012
- GONZÁLEZ-ORTEGA, D. et al. A Kinect-based system for cognitive rehabilitation exercises monitoring. *Computer Methods and Programs in Biomedicine*, v. 113, n. 2, p. 620–631, 2014.
- KALOGIROS, P. KinectJS, Kinect plus Javascript, 2012. Disponível em: <<http://kinect.childnodes.com/docs>>. Acesso em: 16 jun. 2014.
- KAMEL BOULOS, M. et al. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth Navigation. *International Journal of Health Geographics*, v. 10, n. 1, p. 45, 2011.
- LEE, E.; LIU, X.; ZHANG, X. Xdigit: An arithmetic kinect game to enhance math learning experiences. Retrieved February, v. 14, p. 2013, 2012.
- LUTZ, M. Learning python. [S.l.]: “O’Reilly Media, Inc.,” 2013.
- NOYES, B. Smart Client Deployment with ClickOnce: Deploying Windows Forms Applications with ClickOnce. [S.l.]: Pearson Education, 2006.
- PARVINI, F.; SHAHABI, C. An algorithmic approach for static and dynamic gesture recognition utilising mechanical and biomechanical characteristics. *International journal of bioinformatics research and applications*, v. 3, n. 1, p. 4–23, 2007.
- PEDERSOLI, F. et al. XKin -: eXtendable hand pose and gesture recognition library for kinect. *Proceedings of the 20th ACM on Multimedia*. Nara, Japan: ACM., 2012
- SANTANA, O.; GALES, T. Python e Django - Desenvolvimento ágil de aplicações web. São Paulo: Novatec, 2010.
- SILVA, A. J. DA; NETO, E. B. C. Utilização do Kinect como ferramenta de apoio ao processo de ensino e aprendizado. VII CONNEPI - Congresso Norte Nordeste de Pesquisa e Inovação. [S.l.: s.n.]. , 2012.
- SUMA, E. A. et al. FFAST: The Flexible Action and Articulated Skeleton Toolkit. *Virtual Reality Conference (VR)*, 2011 IEEE. [S.l.: s.n.]. , 2011
- SUMA, E. A. et al. Adapting user interfaces for gestural interaction with the flexible action and articulated skeleton toolkit. *Computers & Graphics*, v. 37, n. 3, p. 193–201, 2013.
- VILLAROMAN, N.; ROWE, D.; SWAN, B. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. *Proceedings of the 2011 conference on Information technology education*. West Point, New York, USA: ACM, 2011.