

Construção de um software Media Center com reconhecimento de gestos e comandos de voz utilizando o Microsoft Kinect e os princípios de Natural User Interface

PERNAMBUCO JÚNIOR, I. A. M.

Centro de Ensino Superior de Juiz de Fora, Juiz de Fora, MG

KNOP, I. O.

Resumo: A forma como as pessoas interagem com os computadores sempre foi uma preocupação da Tecnologia da Informação e nas últimas décadas essa interação vem passando por várias mudanças e evoluções. Isso acontece devido à evolução das tecnologias e priorização da facilidade de uso, o que diminui a exigência de conhecimento técnico por parte dos usuários. A Interação Natural se tornou popular com o advento de jogos para videogame com tecnologias como o Kinect e o Wii. No entanto, seu conceito já faz parte da ficção e da literatura científica. Neste trabalho é abordado o desenvolvimento voltado à Interação Natural através de ferramentas que viabilizam o controle e a manipulação de um sistema através de movimentos e comandos de voz, gerados pelo usuário através do uso do Kinect.

Palavras-chave: Natural User Interface; Interface Homem-Máquina; Microsoft Kinect; Computação baseada em gestos.

1 Introdução

Interface é o nome dado a toda a porção de um sistema com a qual um usuário mantém contato ao utilizá-lo, tanto ativa quanto passivamente. A interface engloba tanto software quanto hardware (dispositivos de entrada e saída, tais como: mouse, teclados, monitores, tablets e etc.) (Moran, 1981). Considerando a interação como um processo de comunicação, a interface pode ser vista como o sistema de comunicação utilizado neste processo.

A definição de interface, na área da computação, segundo o dicionário Aurélio (AURÉLIO, 2013) é: “Meio físico ou lógico através do qual um ou mais dispositivos ou sistemas incompatíveis conseguem comunicar-se entre si”. Em um âmbito mais geral, englobando, por exemplo, interfaces entre usuários e veículos ou máquinas industriais, o termo utilizado é interface homem-máquina.

Em linhas gerais, a área de Interação Humano-computador (IHC) investiga o projeto (design), avaliação e implementação de sistemas computacionais interativos para uso humano, com os fenômenos associados a este uso (Hewett et al., 2014). Os estudos relacionados ao projeto de IHC se referem a como construir interfaces de alta qualidade, definindo métodos, modelos e diretrizes. Os estudos relacionados à avaliação de IHC, por sua vez, buscam avaliar a qualidade de um projeto de

interface, tanto ao longo do processo de desenvolvimento como quando o software está pronto.

A interação humano-computador vem cada vez mais sendo evoluída de forma a apresentar ao usuário um uso mais intuitivo, natural e rápido. Portanto, deve-se considerar utilizar recursos disponíveis atualmente para se criar aplicações que utilizam os conceitos de Natural User Interface (NUI).

O presente trabalho apresenta uma discussão sobre a evolução das interfaces em uma perspectiva histórica e um novo sistema de gerenciamento de arquivos multimídia, o Kinect Media Center, desenvolvido como estudo de caso de uma análise do mais recente paradigma da NUI como IHC.

O software desenvolvido, o Kinect Media Center, atua como um *Media Center* para plataforma Microsoft Windows, baseado nos conceitos de NUI, fornecendo suporte para o reconhecimento de gestos e voz, utilizando o hardware do Microsoft Kinect e suas bibliotecas de desenvolvimento. O Kinect é um acessório para o videogame Xbox 360, da Microsoft, lançado em 4 de novembro de 2010 e que permite que os jogadores interajam com os jogos sem ter a necessidade de ter o controle em mãos, ou seja, através dos movimentos e voz.

O aplicativo reconhecerá gestos como, movimentar a mão para a esquerda, para a direita, para baixo e para cima. Adicionalmente, reconhecerá comandos de voz em inglês, como *play*, *next*, *back* e *pause*. Com o propósito de navegar pela interface do software utilizando apenas os gestos e os comandos de voz.

O presente trabalho está organizado em 7 seções. Além desta presente Introdução, a Seção 2 lista os trabalhos relacionados com nossa solução. A Seção 3 abordará a perspectiva histórica da evolução das interfaces homem-máquina. Na Seção 4 falamos sobre o Kinect da Microsoft, sua arquitetura e suas bibliotecas de desenvolvimento. Na Seção 5, falaremos sobre os conceitos do sistema Kinect Media Center, nossa principal contribuição. Na Seção 6 apresentamos as provas do conceito do Kinect Media Center e para finalizar teremos a conclusão na Seção 7.

2 Trabalhos Relacionados

O Kinect foi criado para atuar em conjunto com o videogame Xbox 360, porém rapidamente desenvolvedores começaram a perceber a possibilidade de aplicar a tecnologia presente ao sensor em outras situações. Rapidamente surgiram algumas SDKs para raquear e desenvolver aplicações para o Kinect. Percebendo o interesse

dos desenvolvedores, a Microsoft lançou a versão do Kinect para ambiente Windows, incentivando o desenvolvimento de aplicações para computadores pessoais além de jogos. Atualmente, diversas empresas vêm mostrando interesse em experimentar as qualidades do dispositivo.

Entre os casos de destaque, está o Hospital Evangélico de Londrina (PR), que em parceria com o grupo Amil, criou um aplicativo cuja função é auxiliar o trabalho de médicos e enfermeiros nas salas de cirurgia. O aplicativo utiliza os recursos de reconhecimento de gestos do Kinect para que os profissionais da saúde possam acessar o banco de dados da instituição e visualizar arquivos digitais, imagens, exames e vídeos, assim como a ficha completa do paciente, apenas com gestos das mãos à distância, em uma TV ligada ao acessório dentro do centro cirúrgico conforme pode ser visto na Figura 1.

Figura 1



Sistema Intera - Kinect na sala de cirurgia
Fonte: MORALES, 2012.

O Kinect é usado para que o cirurgião não tenha que tocar os controles do computador para ajustar imagens e visualizar exames. Dessa forma se ganha tempo, uma vez que assim não é necessário ficar realizando todo o processo de esterilização a cada parada (MORALES, 2012).

Em nossa solução, o Kinect Media Center é um sistema multimídia capaz de navegar pelo conteúdo da aplicação através de gestos e comandos de voz. O usuário consegue visualizar arquivos de imagens, músicas e vídeos, assim como a sinopse de cada filme, apenas com gestos das mãos ou com comandos de voz, de forma bem natural.

Nas pesquisas realizadas foram descobertas e testadas várias aplicações “plug-ins” para sistemas de Media Center, como o plug-in para Windows Media

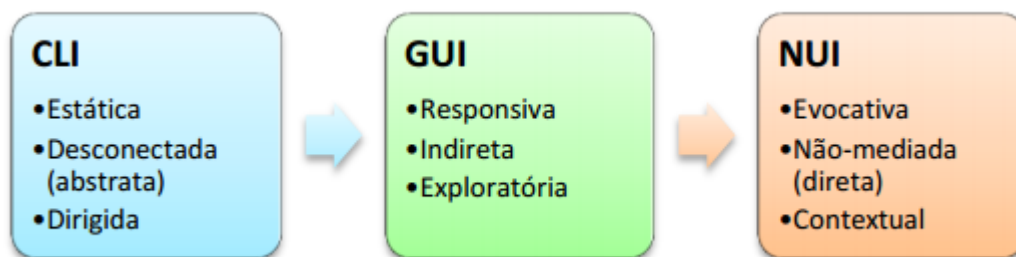
Center (MAUK, 2013) e o plug-in para XBMC (KINEMOTE, 2014), porém nenhum com interface própria e adequada ao uso da NUI. Como são aplicações desenvolvidas para uma interface gráfica, os botões e quadros de imagem são muito pequenos para uma interação gestual, o que pode causar uma experiência de uso desagradável ao usuário.

3 Evolução das Interfaces Homem-Máquina

Um dos aspectos fundamentais do desenvolvimento de sistemas para computador é o projeto de interface com o usuário; ele é fundamental para o sucesso de um sistema. A interface é um meio de interação entre o homem e o computador. É através dela que o usuário consegue manipular todas as informações necessárias durante seu trabalho. “A interface é, de muitas maneiras, a embalagem do software de computador” (PRESSMAN, 1995, p.602).

Na evolução das interfaces homem-máquina é possível identificar três tipos diferentes: Command-Line Interface (CLI) ou Interface de Linha de Comando, Graphical User Interface (GUI) ou Interface Gráfica do Usuário e Natural User Interface (NUI) ou Interface Natural do Usuário, conforme destacado na Figura 2.

Figura 2
Evolução das interfaces de usuário



Fonte: Do autor.

3.1 Command Line Interface (CLI)

A partir de 1970, os *mainframes*, que eram computadores de grande porte com alta escala de processamento, começaram a adotar interfaces que permitiam interatividade com o usuário. A interação por linha de comando substituiu a chamada computação em lote. No processamento por lote as pessoas não podiam interagir

em tempo real com uma máquina do tipo lote, o processo realizava todos os processamentos sem intervenção manual até ser completado.

Com a transição de lote para CLI, o modelo de interação tornou-se baseado em uma série de transações de pedidos e respostas expressos em um vocabulário especializado. As respostas às requisições tornaram-se mais rápidas, quase em tempo real. Entretanto, ainda existia um problema: utilizar uma interface de linha de comando exigia grande tempo e esforço para aprendizagem dos comandos disponíveis. Para solucionar esse problema surgiu a GUI (LANDLEY; RAYMOND, 2004).

3.2 Graphical User Interface (GUI)

Uma *Graphical User Interface* (GUI) é um tipo de interface de usuário que permite a interação com dispositivos digitais através de elementos gráficos como ícones, menus e outros indicadores visuais. A interação é realizada a partir de um dispositivo físico de entrada que controla a posição de um cursor, um mouse, e apresenta a informação organizada em janelas e através de ícones que a representem. Os comandos são dispostos em menus para serem então selecionados e acionados através do mouse (GARBIN, 2010). Deste modo não é necessário memorizar diversos comandos, como é o caso da CLI. Com um ambiente mais amigável, torna-se mais fácil a utilização do sistema operacional, a realização de operações multitarefas, além de permitir a utilização de aplicações como os *widgets*¹.

Com a interface gráfica de usuário os computadores pessoais tornaram-se uma ferramenta de uso mais simples, facilitando o uso, mesmo para pessoas sem amplo conhecimento técnico e iniciantes, o que de fato aumentou a curva de aprendizagem.

3.3 Natural User Interface (NUI)

Interface Natural do Usuário ou NUI (do inglês, Natural User Interface) é o conceito comum utilizada por designers e desenvolvedores de interfaces de

¹ Um widget é um componente de uma interface gráfica do usuário (GUI), o que inclui janelas, botões, menus, ícones, barras de rolagem, etc. Outro emprego do termo são os widgets da área de trabalho, pequenos aplicativos que flutuam pela área de trabalho e fornecem funcionalidade específicas ao utilizador (previsão do tempo, cotação de moedas, relógio, etc.).

computador para se referir a uma interface de usuário que se torna invisível para seus usuários após sucessivas interações. Isso significa que uma interface natural pode ser operada através de ações intuitivas relacionadas aos comportamentos naturais do cotidiano ou de outros contextos, dependendo do propósito e dos requisitos do usuário (BUXTON, 2010).

A palavra natural é utilizada porque a maioria das interfaces usam dispositivos de controle artificial cuja operação deve ser aprendida por meio de botões, como um controle de videogame.

Blake (2010) faz uma comparação entre NUI, CLI e GUI:

Uma CLI utiliza texto como *input* e também como *output* e uma GUI utiliza, por exemplo, mouse para *input* e janelas, menus, ícones como *output*. Porém, a diferença principal, é o fato de CLI e GUI se definirem quanto ao dispositivo de entrada utilizado, enquanto NUI se define no modo como é realizada a interação.

A Interação Natural (NI, ou Natural Interaction) é um conceito que estuda formas do homem interagir com dispositivos eletrônicos através dos sentidos do ser humano, seja através de gestos, comandos de voz, movimentos e expressões corporais ou detecção e identificação de partes do corpo humano como rosto, mão, polegar, retina, articulações, entre outros.

O seu propósito é tornar a experiência do usuário mais próxima de contextos e diálogos do cotidiano, sem a necessidade de aprendizado de novos procedimentos, comandos, linguagens ou dispositivos eletrônicos. O controle de aplicações e sistema pode ser feito de diversas maneiras, entre elas, por toque em telas touch, reconhecimento de gestos, de posições específicas, detecção da face do usuário ou até mesmo do som emitido por ele.

O foco deste trabalho é desenvolver um software capaz de ser manipulado apenas com gestos e comandos de voz, de forma rápida e intuitiva. Apresentando todas as características de uma interação mais natural.

4 Microsoft Kinect

O Kinect foi lançado em 4 de novembro de 2010 e foi um grande marco na indústria de jogos. O aparelho é um sensor de movimentos desenvolvido para o Xbox 360, junto com a empresa Prime Sense. O Kinect criou uma nova tecnologia

capaz de permitir aos jogadores interagir com os jogos eletrônicos sem a necessidade de ter em mãos um controle (do inglês, *joystick*), inovando no campo da jogabilidade.

A ideia por trás do Kinect foi exatamente explorar ao máximo os conceitos da NUI, ao retirar do jogador qualquer tipo de controle, e inseri-lo dentro do game, de forma natural, usando os movimentos do seu próprio corpo, com isso o Kinect trouxe uma imersão jamais vista antes no mundo dos games.

O Kinect foi um grande sucesso de vendas e entrou para o Guinness Book como o aparelho eletrônico de vendagem mais rápida na história da indústria (FASTEST-SELLING... 2011), o aparelho vendeu 2,5 milhões em apenas 30 dias, ou seja, uma média de 100 mil unidades por dia, ao redor do mundo.

4.1 Microsoft Kinect - Arquitetura e SDK

Existem algumas especificações e características relevantes da arquitetura do Kinect (Figura 3), que devem ser levadas em consideração ao se desenvolver uma aplicação. Dentre eles estão os principais componentes de hardware que compõem o Kinect, que são:

- **Câmera RGB:** Utilizada para reconhecimento facial e detecção de outras características por meio do reconhecimento de três componentes de cor: vermelho, verde e azul;
- **Sensores de profundidade:** Possui um projetor de luz infravermelha e um sensor CMOS (Complimentary Metal-Oxide Semiconductor) que funcionam juntos para identificar a profundidade dos objetos na sala independentemente das condições de iluminação;
- **Microfone multi-array:** Um conjunto de quatro microfones capaz de isolar as vozes dos jogadores de outros barulhos dentro da sala. Isso permite que o jogador possa se manter mais distante do sensor e ainda usar controles por voz.
- **Motor de inclinação:** O motor de inclinação serve para ajustes do sensor, tem uma inclinação de 27° para cima ou para baixo, para o cálculo de rastreamento da distância do corpo.

Figura 3
Componentes do sensor Kinect.



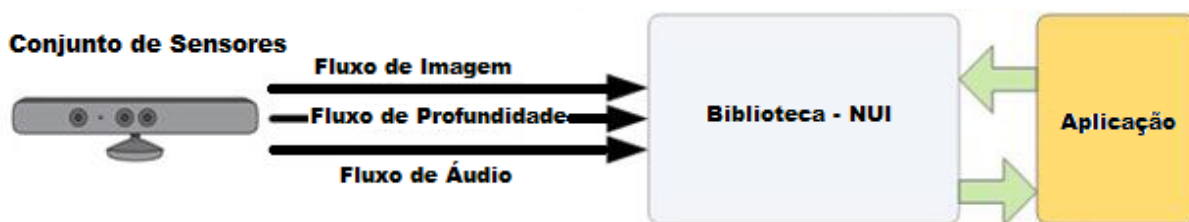
Fonte: PAULA, 2011.

As especificações técnicas do Kinect indicam informações que nos ajudam a perceber o funcionamento do mesmo. Outros fatores importantes são a resolução máxima do sensor de 640x480 pixels, rodando a 30 FPS (do inglês, *frames per second* ou quadros por segundo). O ângulo de visão do Kinect (motor de inclinação) é de 57,5° na horizontal e 43,5° na vertical, a partir de +27° a -27°. O Kinect pode ser utilizado a diferentes distâncias, dependendo do modo escolhido para o sensor de profundidade. Existem dois modos para escaneamento: o Modo Próximo (*Near Mode*) que pode manipular objetos entre 40 cm e 3 metros e o Modo Padrão (*Default Mode*), que pode manipular objetos entre 80 cm e 4 metros.

Atualmente existem duas versões do Kinect, a versão para Xbox 360 e a versão para Windows. Neste trabalho utilizaremos o SDK para Windows, e o Kinect para Xbox 360. Existem algumas incompatibilidades quando se utiliza o Kinect para Xbox, algumas funções da SDK podem gerar erros se utilizados com a versão do Kinect para Xbox (como o controle dos parâmetros da câmera e mapeamento de face).

A versão atual da SDK é a 1.8 (KINECT... 2013), que nos fornece o rastreamento de todas as partes do corpo, incluindo todas as informações capturadas pelo sensor, como todos os fluxos retornados por ele, como pode ser visto na Figura 4.

Figura 4
Integração entre hardware e software com a aplicação.

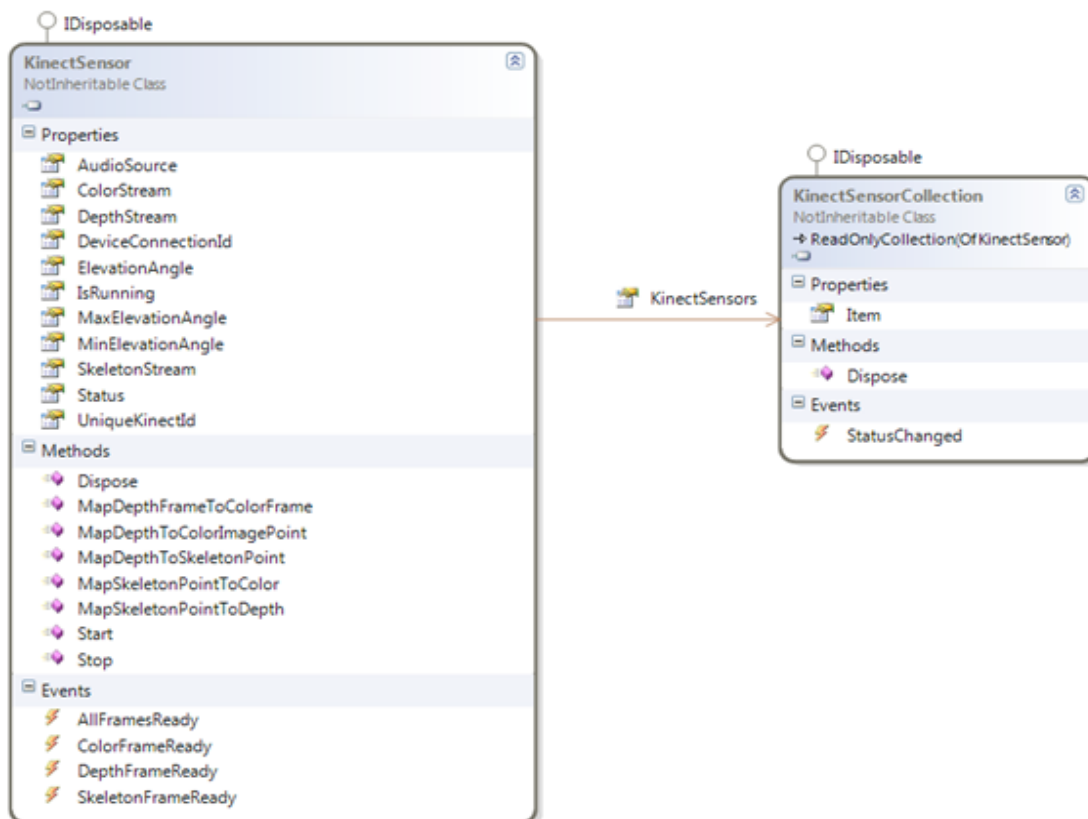


Fonte: CATURRE, 2011

O Kinect retorna três fluxos de dados, o *Image Stream* (fluxo de imagens), o *Depth Stream* (fluxo de profundidade) e o *Audio Stream* (fluxo de áudio). Através das classes da API, conseguimos capturar todos os fluxos. Abaixo estão os principais diagramas de classe utilizados no desenvolvimento da aplicação.

A classe Kinect Sensor fornece acesso as funcionalidades do Kinect, devemos recuperar a instância do `KinectSensor` a partir da coleção `KinectSensor` como visto na Figura 5.

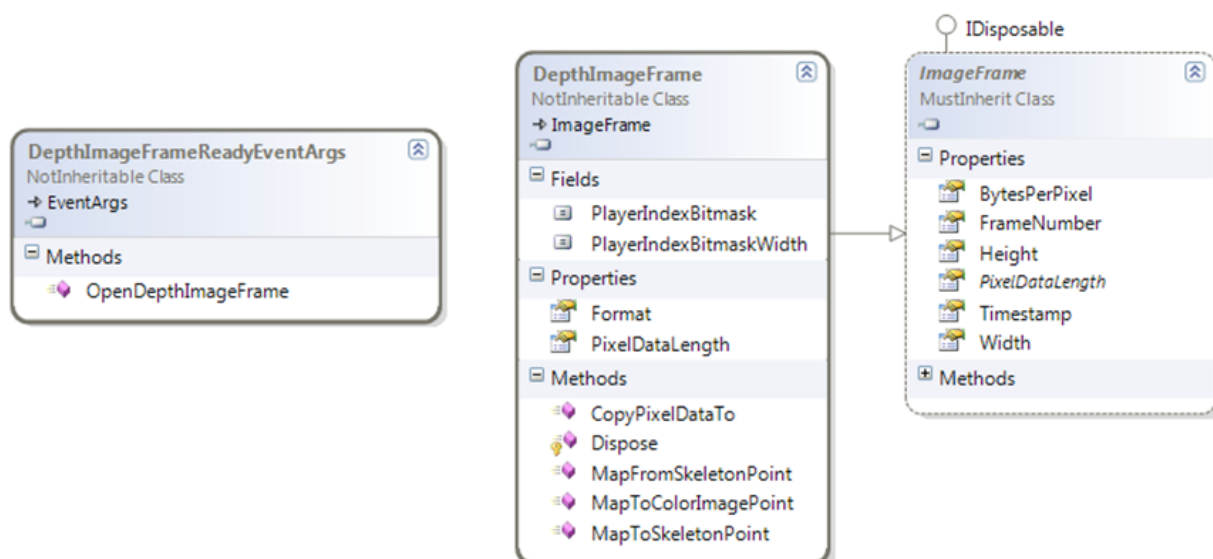
Figura 5
Kinect SDK - Classe Kinect Sensor.



Fonte: Do autor

Para os dados de profundidade a classe `KinectSensor` nos fornece o evento `DepthFrameReady`, para a gestão do quadro de profundidade (Figura 6). O `depth` retorna pixels que contêm a distância em milímetros a partir do plano da câmera e o jogador. Cada ponto que o `depth` retorna a profundidade do objeto em relação ao Kinect (em milímetros), além do número do jogador (se for parte de algum). Os objetos mais perto do Kinect vão ficando pretos enquanto mais longe vão ficando brancos. Isso só é possível porque cada ponto (*pixel*) desses tem uma informação de posicionamento em três dimensões, nos eixos x, y e z.

Figura 6
Kinect SDK - DepthImageFrameReadyEventArgs .



Fonte: Do autor

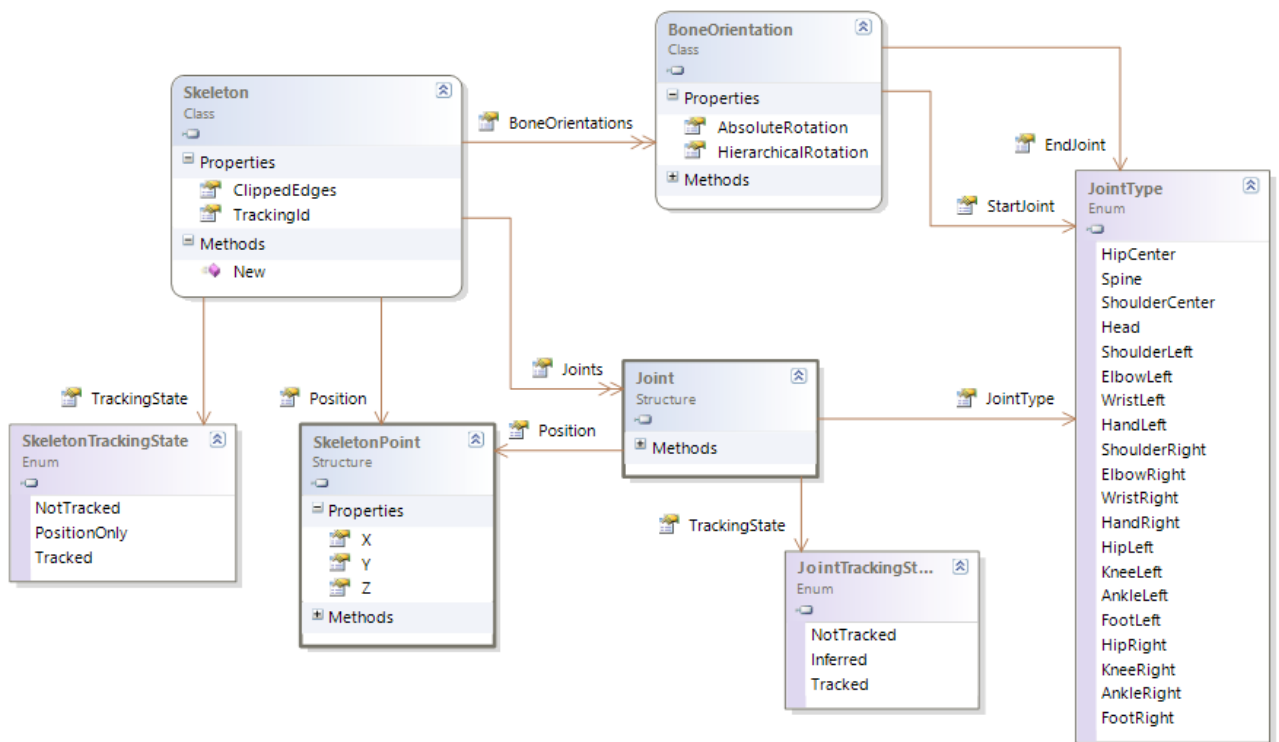
Uma das classes centrais da SDK é a classe `Skeleton` (Figura 7). A câmera do Kinect é equipada com um chip desenvolvido para comparar as informações extraídas das imagens com partes do corpo humano já conhecidas e, em seguida, essas partes identificadas são utilizadas para estimar a pose do usuário, fornecendo informações sobre o esqueleto identificado, como as diferentes articulações (Shotton, J. et al., 2011).

O esqueleto fornecido pelo sensor do Kinect contém um conjunto de até vinte diferentes articulações identificadas, cada uma dessas articulações representa uma articulação correspondente no corpo humano. Cada jogador possui um conjunto de

todas essas articulações com suas posições em três eixos diferentes no espaço, identificados por x, y e z. Com o eixo z representando o quão perto ou longe a pessoa está do sensor e distâncias medidas em metros. Cada articulação possui um estado associado:

- **Tracked:** Se a articulação está visível ao sensor;
- **Not Tracked:** Se a articulação está fora do alcance do sensor;
- **Inferred:** Se a articulação não está visível mas pode ser inferida por meio de interpolação entre dados das articulações adjacentes.

Figura 7
Kinect SDK - Classe Skeleton.



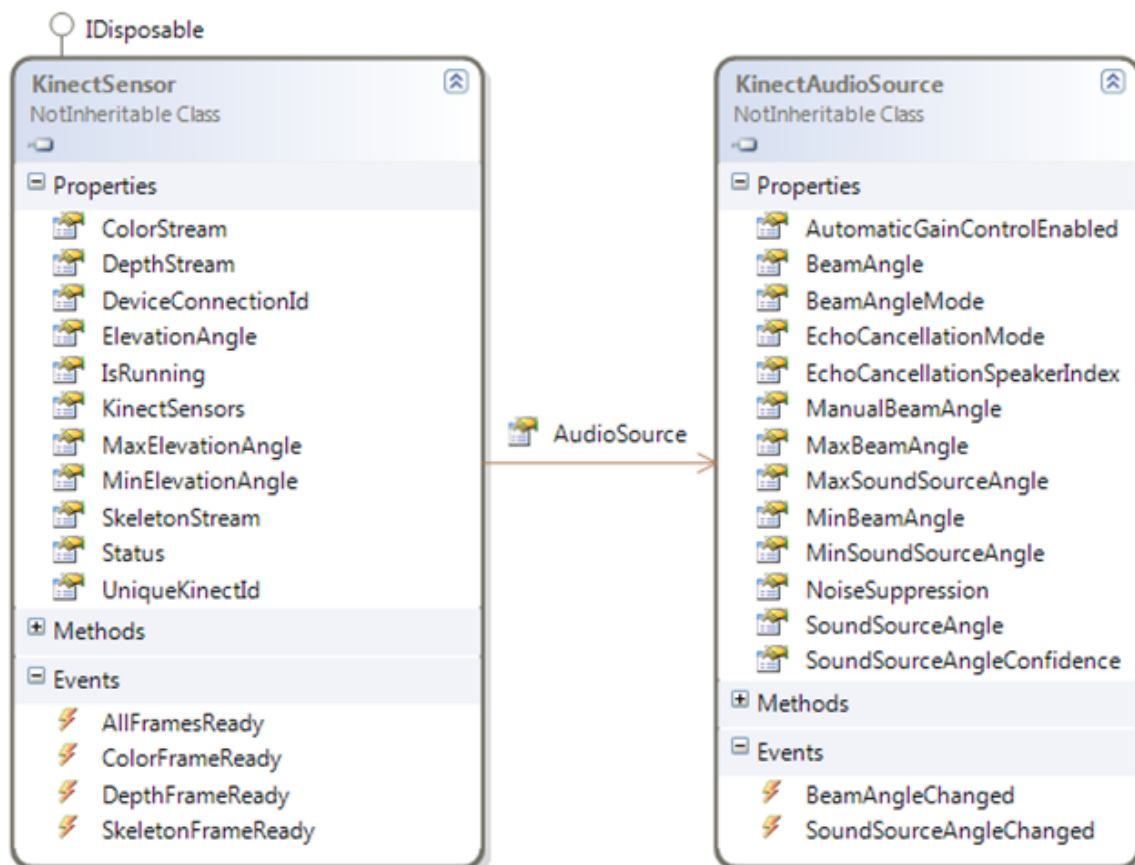
Fonte: Do autor

O Kinect para Windows SDK também é capaz de ser uma fonte para o *Microsoft Speech API*. Portanto é possível fazer o reconhecimento de voz utilizando o Kinect.

O Kinect vem com um grupo de quatro microfones com a funcionalidade de “processamento de áudio”, que captura som de alta qualidade. O sensor está equipado com um processador digital de sinais (DSP do inglês Digital Signal Processor), que é usado para remover o ruído de fundo e cancelar efeitos de eco. A

classe `KinectSensor` expõe as propriedades `AudioSource` do tipo `KinectAudioSource` através do qual você pode gerenciar todos os recursos de áudio como pode ser vista na Figura 8.

Figura 8
Kinect SDK - Classe KinectAudioSource.



Fonte: Do autor

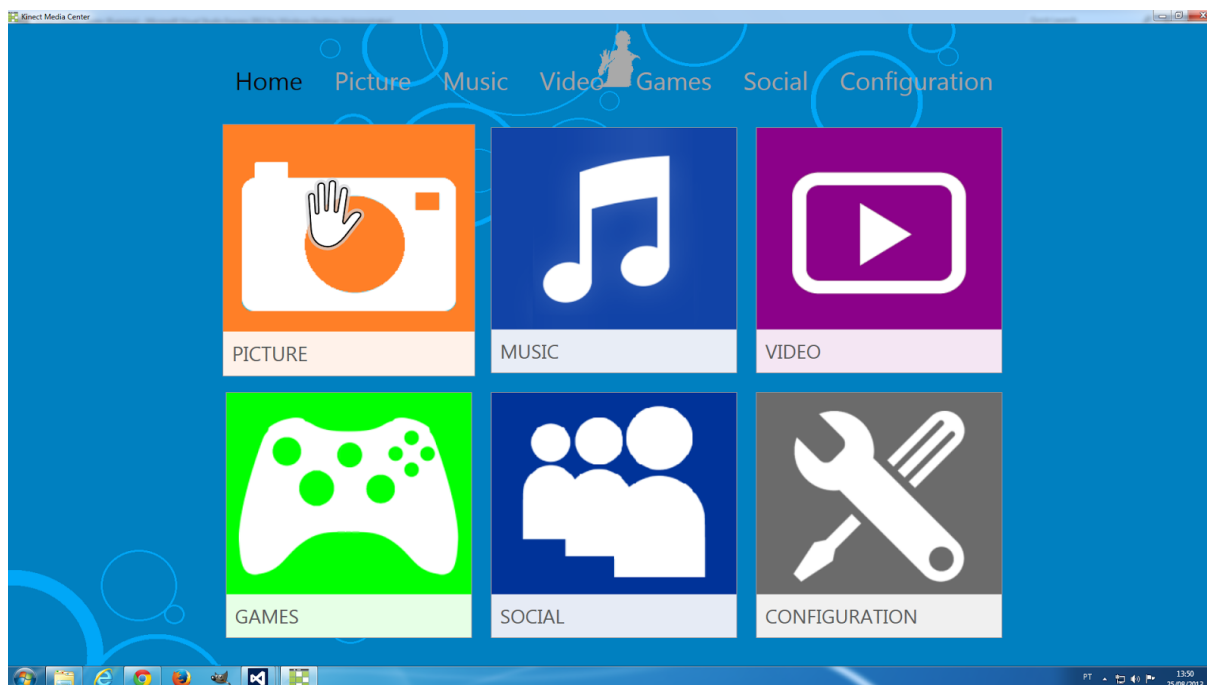
5 Kinect Media Center

Foi desenvolvido um protótipo de um sistema multimídia, o Kinect Media Center, composto por hardware e software que permita o estudo da integração de elementos de Interfaces de Usuário Naturais ao acionamento de eventos da aplicação através do uso das técnicas referentes à NUI.

Nos dias de hoje, as pessoas estão transformando suas salas de estar em verdadeiros ambientes multimídias, onde pode-se visualizar fotos, vídeos, músicas, internet, TV, games, tudo em um ambiente integrado. O Kinect Media Center propõe aumentar a consciência situacional do usuário, permitindo que através de uma

representação virtual de um ambiente, um usuário seja capaz de operar o sistema de forma natural, utilizando apenas gestos e comandos de voz, para controlar todo o conteúdo de sua biblioteca multimídia, conforme a Figura 9.

Figura 9



Kinect Media Center - Tela Inicial do Sistema

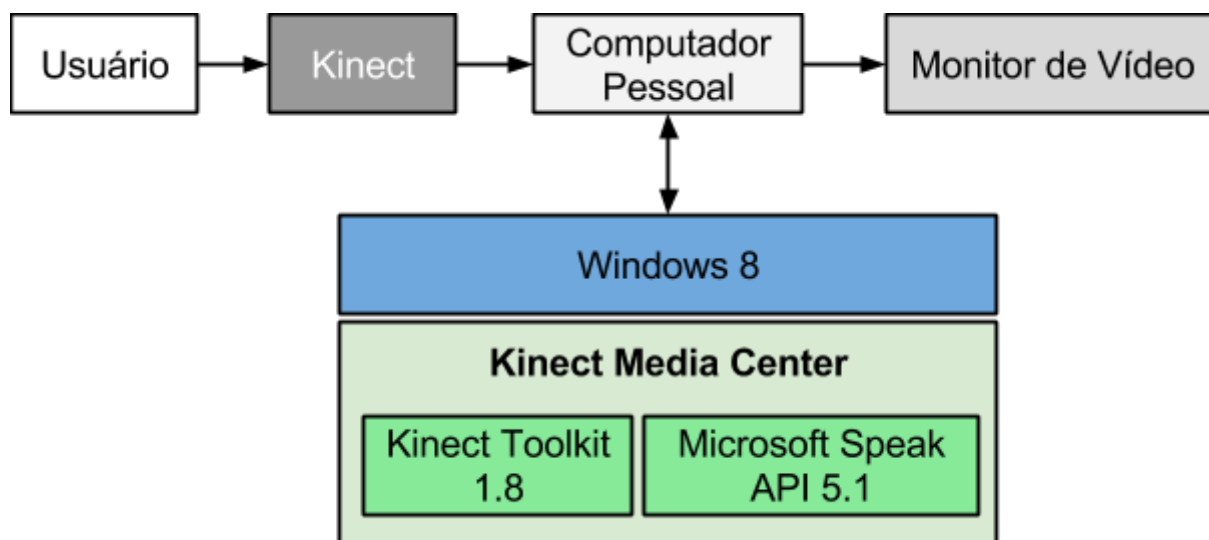
Fonte: Do autor.

Para a Interface Natural, decidiu-se utilizar uma abordagem baseada em Visão Computacional, utilizando como sensor uma câmera disponível comercialmente. O Kinect é um equipamento disponível no comércio com um custo relativamente baixo. O sensor Kinect foi utilizado para captura do movimento do usuário e para os controles acionados por voz, para uma interação natural com o sistema.

A aplicação não utiliza banco de dados, o software tem o funcionamento parecido com os softwares embarcados em Smart TVs da Samsung, como o modelo F8000. Os diretórios são pré-definidos e o usuário apenas navega pelo conteúdo.

A arquitetura da solução utiliza o sensor Kinect ligado a um microcomputador, onde estão instalados os *drivers* e as bibliotecas de desenvolvimento conforme o diagrama esquemático da Figura 10.

Figura 10
Diagrama de Arquitetura do Kinect Media Center



Fonte: Do Autor

O movimento do usuário será captado pelo Kinect e processado pela aplicação, a qual realizará a extração do gesto do usuário e acionará o evento correspondente no ambiente.

O sistema foi desenvolvido utilizando o Visual Studio 2013 Express, a linguagem C# e a plataforma .NET 4.5. C# é uma linguagem de programação, de tipos protegidos, orientada a objeto e que permite aos desenvolvedores construir uma variedade de aplicações seguras e robustas, compatíveis com o .NET Framework.

Para a criação da camada de visualização, foi utilizada a tecnologia Windows Presentation Foundation (WPF), elaborada pela Microsoft para o desenvolvimento de aplicações com interfaces ricas. Além do desenho de formulário simples, ela também permite o desenho de objetos 2D e 3D. Como API de comunicação com o Kinect foi usado o SDK oficial da Microsoft que atualmente está na versão 1.8.

A utilização do WPF mostrou-se acertada. A interação com o usuário está sofrendo uma revolução, o próprio Kinect é um exemplo disso, além disso, a experiência do usuário é uma área de interação homem computador que vem ganhando destaque nos últimos meses com o lançamento do Windows 8 e sua nova interface Metro UI, por isso, este projeto procurou aplicar os conceitos de Metro UI e o WPF mostrou-se a ferramenta ideal para isso. Alguns dos conceitos da interface Metro UI são:

- **Conteúdo em primeiro lugar:** O conteúdo é a parte central dos aplicativos estilo Metro, e colocar o conteúdo em primeiro lugar é fundamental para o design de aplicativos com este estilo.
- **Rápido e fluido:** As interações com o usuário e transições são rápidas e intuitivas, e as animações são planejadas e intencionais.
- **Simples e bonito:** O sistema precisa ser simples, controles adicionais como painéis de alto-relevo, caixas de agrupamento devem ser dispensadas. O sistema precisa ser bonito e limpo, utilizando uma tipografia adequada.

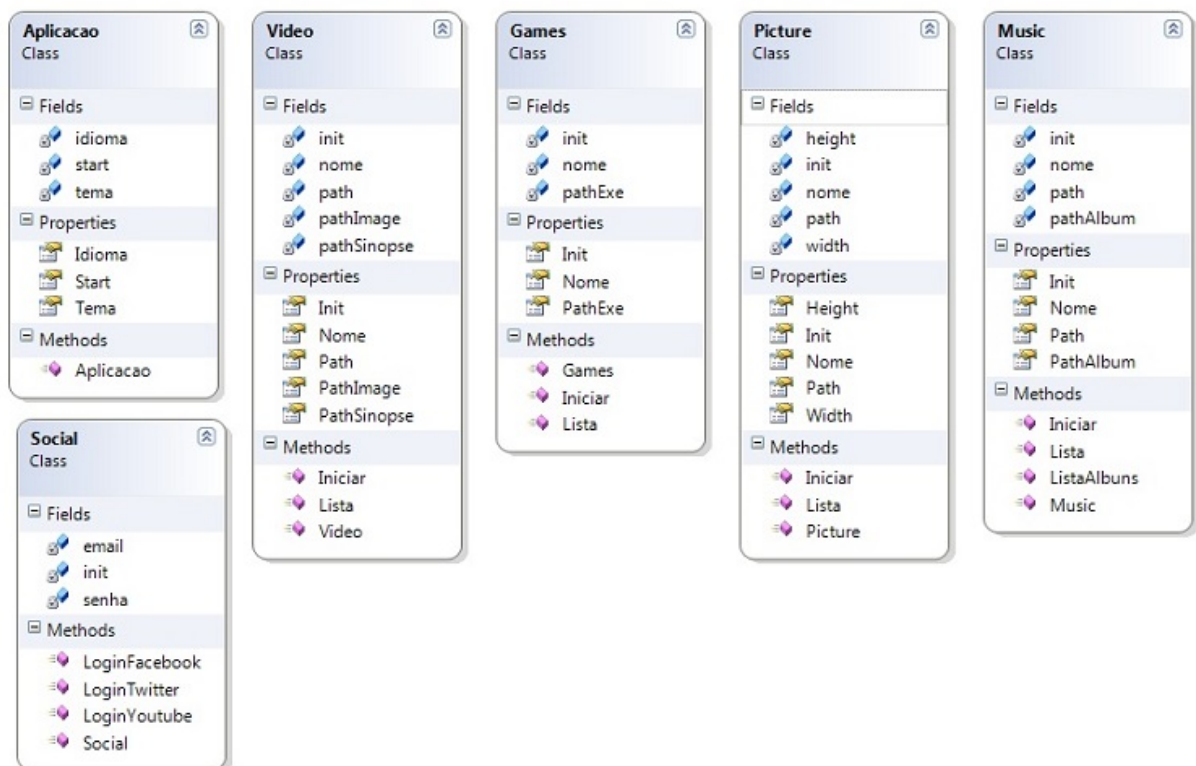
Os conceitos que a linguagem visual do Metro UI são particularmente interessantes para softwares desenvolvidos com grande apelo visual, como o desenvolvido neste trabalho. Dado que o resultado final necessitava de uma interface limpa, fácil e amigável, consideramos que a adoção da Metro UI foi acertada.

6 Prova de Conceito

Durante o desenvolvimento, foram geradas várias versões do software, sempre aprimorando a funcionalidade do mesmo. Chamado de Kinect Media Center, a versão final é capaz de reconhecer gestos e comandos de voz, estes utilizados para navegar pelas imagens de um diretório, controlar a execução de músicas ou de vídeos de uma biblioteca.

A seção de software responsável pelo reconhecimento de gestos foi obtida através do SDK 1.8 oficial da Microsoft. O software foi desenvolvido com código fonte em C# e WPF. O diagrama de classe do sistema Kinect Media Center mostra as principais classes do sistema, e nele não há relacionamentos ou heranças como pode ser visto na Figura 11.

Figura 11
Diagrama de Classe - Kinect Media Center

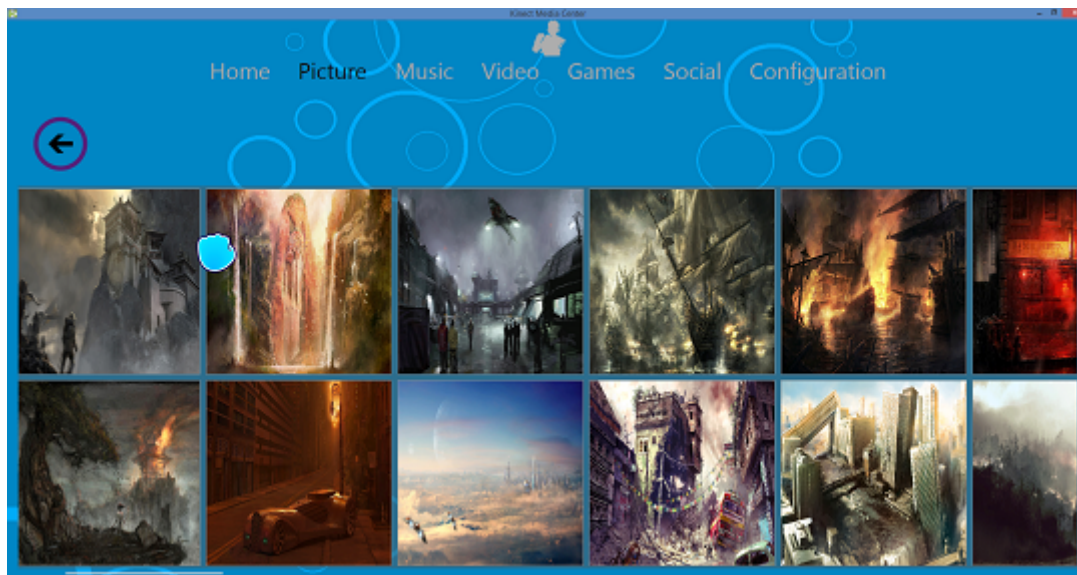


Fonte: Do autor.

Para a utilização do sistema o usuário deve iniciar o aplicativo, em seguida se posicionar em frente ao sensor Kinect, o sistema chama o método de escaneamento do Kinect, instanciando a classe `KinectSensor`. O Kinect retorna o fluxo de imagem, uma caixa no topo, ao centro, mostra o usuário escaneado e um ponteiro com um ícone de uma mão é mostrada na tela quando o usuário levanta uma das mãos.

Para navegação do sistema basta mover a mão em direção ao item desejado, o ponteiro do sistema move-se como o ponteiro do mouse, para clicar basta fazer o movimento de aproximar a mão, como um clique, para todos os gestos de navegação foram utilizados a classe `Skeleton`. A Figura 12 mostra a tela de navegação pela biblioteca de imagens do sistema.

Figura 12
Kinect Media Center - Biblioteca de Imagens.



Fonte: Do Autor.

Para os controles de voz, a cada tela do sistema, é instanciado um dicionário de palavras que podem ser reconhecidas, geralmente as palavras são indicadas na tela do sistema, e cada tela tem um dicionário próprio. A API do Microsoft Speech não contém um pacote de linguagens em português, portanto os termos para reconhecimento de palavras devem ser em inglês. A classe `KinectAudioSource` foi utilizada para o método de reconhecimento de comandos de voz.

Para fornecer um retorno adequado ao usuário, existem alguns controles que serão dispostos na tela do aplicativo para fornecer o feedback. Abaixo esta a tabela com as indicações dos feedbacks para controle (Tabela 1):

TABELA 1**Ícones de feedback ao usuário do Sistema Kinect Media Center**

			
Cursor padrão (mão esquerda)	Cursor sobre algo que é acionável (botão)	Indicador de progresso (cor preenche mão conforme o usuário pressiona)	Estado totalmente pressionado
			
Cursor mover (Gripped)	Cursor padrão (mão direita)	User Viewer (indica ao usuário que ele está sendo visto pelo sensor)	Indicador de controle por voz.

Fonte: Do Autor

7 Conclusão

Com a evolução das tecnologias, vem surgindo diversas formas de manipular e interagir com computadores, jogos e até mesmo instrumentos de trabalho. O Kinect surgiu com uma ótima proposta para executar e interagir com esses mecanismos. Porém, nota-se que essas interfaces devem apresentar características ímpares às encontradas nas interfaces tradicionais para que possam ser consideradas, de fato, naturais e que permitam uma interação mais fluida e intuitiva.

Este trabalho foi desenvolvido com foco nas tecnologias que permitem o uso da interação natural do homem com o computador utilizando reconhecimento de voz e gestos. Foi desenvolvido um aplicativo de gestão multimídia para o sistema Windows que utiliza o hardware do Kinect, ambos da Microsoft.

O trabalho também incluiu uma revisão das tecnologias de desenvolvimento de interfaces que evoluíram para a Interação Natural. O ambiente de desenvolvimento do Kinect foi escolhido como tecnologia para a implementação da NUI, devido ao seu baixo custo. Atrás de suas ferramentas de desenvolvimento conseguimos capturar comandos de voz em inglês e gestos pela câmera.

O presente trabalho não esgota todas as possibilidades de uso da arquitetura e pode ser estendido posteriormente em trabalhos futuros. Como melhoria direta, podemos destacar a necessidade de se incluir comandos em português, pois o Microsoft Speech API não fornece esta linguagem por padrão.

Uma outra melhoria é a utilização de gestos mais complexos, como uso das duas mãos na tela de visualização de imagens para controle de zoom. A integração de um sistema de banco de dados pode acrescentar a funcionalidade de anotações e preferências do usuário. Por fim, a integração com um navegador para Internet interno ao sistema, capaz de navegar pelas redes sociais através de gestos.

Referências

AURÉLIO, On Line. **Significado de Interface**, Disponível em: <<http://www.dicionariodoaurelio.com/Interface.html>>. Acesso em: 4 de abril. 2014.

BLAKE, J. Multi-touch Development with. **WPF and Silverlight**. Greenwich, Connecticut, EUA: Manning Publications, 2010.

BUXTON, B. CES 2010: NUI with Bill Buxton. **Channel 9 Blog for Larry Larsen**. Disponível em: <<http://channel9.msdn.com/blogs/larrylarsen/ces-2010-nui-with-bill-buxton>>. Acesso em: 15 abril. 2014.

CATURRE, David. **Unleash the power of Kinect for Windows SDK!, 2011**. Disponível em: <<http://blogs.msdn.com/b/eternalcoding/archive/2011/06/13/unleash-the-power-of-kinect-for-windows-sdk.aspx>>. Acesso em: 5 maio. 2014.

FASTEST-SELLING gaming peripheral. 2011. Disponível em: <<http://www.guinnessworldrecords.com/records-9000/fastest-selling-gaming-peripheral/>>. Acesso em: 21 jun. 2014.

GARBIN, S. M. **Estudo da evolução das interfaces homem-computador**. 2010. 90 f. Monografia (Graduação) – Curso de Engenharia Elétrica, Departamento de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010. Cap. 4. Disponível em: <www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-25112011-104445/publico/Garbin_Sander_Maeda.pdf>. Acesso em: 7 set. 2012.

Hewett, T.; Baecker, R.; Card, S.; Carey, T.; Gasen, J.; Mantei, M.; Perlman, G.; Strong, G.; Verplank, W. (1992) “ACM SIGCHI **Curricula for Human-Computer Interaction**”. ACM SIGCHI Report, ACM, NY. Disponível em <<http://sigchi.org/cdg/>> Acesso em: 12 out. 2013.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp e M. Finocchio, “**Real-Time Human Pose Recognition in Parts from Single Depth Images**,” em IEEE Computer Vision and Pattern Recognition, CVPR, Colorado, 2011.

MAUK, Brad. **Kinect for Media Center**. 2013. Disponível em: <<http://www.themediacenterproject.com/KinectForMediaCenter>>. Acesso em: 12 abr. 2014.

KINECT for Windows SDK: Downloads. 2013. Disponível em: <<http://www.microsoft.com/en-us/kinectforwindowsdev/Downloads.aspx>>. Acesso em: 20 mar. 2014.

KINEMOTE: Gesture Driven Control. Disponível em: <<http://www.kinemote.net/>>. Acesso em: 12 abr. 2014.

LANDLEY, Rob; RAYMOND Eric Steven. **The Art of Unix Usability**, Disponível em: <<http://www.catb.org/~esr/writings/taouu/html/index.html>>. Acesso em: 4 de ago. 2012.

MORALES, Fernanda; **Hospital brasileiro usa o Kinect durante cirurgias para visualização de exames**. Disponível em: <<http://corporate.canaltech.com.br/noticia/saude/Hospital-brasileiro-usa-Kinect-da-Microsoft-durante-cirurgias/>>. Acesso em maio de 2014.

Moran, T. P. (1981). The command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies*, 15, 1, 3-50.

PAULA, Bruno C. de. **Adaptando e Desenvolvendo jogos para uso com o Microsoft Kinect**: SBGames 2011. 2011. Disponível em: <<http://www.brunocampagnolo.com/tutorialkinect2011/>>. Acesso em: 12 maio 2014.

PRESSMAN, Roger S. **Engenharia de software**. Tradução de José Carlos Barbosa dos Santos; revisão técnica José Carlos Maldonado, Paulo César Masieiro, Rosely Sanches. São Paulo: Makron Books 1995. Pag. 1056. Título original: Software engineering.