

**CENTRO UNIVERSITÁRIO NOSSA SENHORA DO PATROCÍNIO
FACULDADE DE INFORMÁTICA
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

Interface para apresentações interativas com Kinect/NUI

LEONARDO BOLOGNESI

**ITU - SP
2014**

LEONARDO BOLOGNESI N.25 RGM: 088034

Interface para apresentações interativas com Kinect/NUI

**TCC apresentado ao Centro
Universitário Nossa Senhora do
Patrocínio como parte dos requisitos
para a conclusão do curso de análise e
desenvolvimento de sistemas.**

Orientador(a): Prof. Carlos Alberto

**ITU – SP
2014**

O Centro Universitário Nossa Senhora do Patrocínio não se responsabiliza pelas informações contidas neste Trabalho Essas informações são de responsabilidade exclusiva do autor.

RESUMO

O desenvolvimento de software utilizando os conceitos de *Natural User Interface* (NUI), traz grandes inovações afinal é uma nova forma de se pensar em interação homem máquina, a chave desta interface é o aproveitamento das habilidades e recursos naturais já existentes, trazendo as aplicações para um nível maior de inteligência, afinal o software pode reconhecer os atributos naturais e entender o que o usuário quer, chegando a um nível cognitivo de interface tratando os utilizadores de forma singular e personalizada, mas apenas os conceitos de interface NUI (interação utilizando gestos, movimentos e voz detalhada no capítulo 2.1 deste trabalho) não possibilitam a materialização de um software, existe a necessidade de um hardware que possa capturar as informações vindas do mundo real, e justamente a proposta de NUI da Microsoft é o sensor de movimentos Kinect, hardware especificamente desenvolvido para NUI, lançado primeiramente para os consoles Xbox 360 e logo disponibilizado para desenvolvimento em PC's. Este projeto tem por objetivo o desenvolvimento introdutório utilizando o sensor de movimento Kinect e seu Software Development Kit (SDK) oficial, para interagir com apresentações feitas em PowerPoint e a utilização destes recursos em uma tela projetada, visando facilitar a interação do apresentador tornando a apresentação algo natural e dinâmico.

Palavras-chave: Kinect, Natural User Interface, apresentações PowerPoint.

Lista de Ilustrações

Figura 1: Hardware do Kinect.....	12
Figura 2: Kinect área de cobertura.....	13
Figura 3: Microsoft Kinect SDK.....	13
Figura 4: Power Point Versão 2.00.....	14
Figura 5: Exemplo do KinectTileButton Utilizado na aplicação.....	17
Figura 6: Exemplo de execução síncrona.....	18
Figura 7: Exemplo de Execução Assíncrona.....	18
Figura 8: Microsoft.Kinect.....	19
Figura 9: Tela para seleção de Apresentações.....	20
Figura 10: Fluxo interação KinectRegion.....	21
Figura 11: Status Kinect- Reconhecido.....	21
Figura 12: Status Kinect - Configurando.....	21
Figura 13: Status Kinect - Desconectado.....	21
Figura 14: Interação da Tela Principal.....	22
Figura 15: Juntas do esqueleto - IKF.....	23
Figura 16: Kinect - Juntas do esqueleto.....	23
Figura 17: Fluxo completo de interações.....	24
Figura 18: IKF- Cronograma do Projeto.....	33
Figura 19: Diagrama Caso de Uso - Principal.....	34
Figura 20: Diagrama caso de uso - Manter arquivos Apresentação.....	35
Figura 21: Diagrama caso de uso - Iniciar Apresentação.....	35
Figura 22: Diagrama de Classes.....	36
Figura 23: Diagrama de Sequência.....	37
Figura 24: Diagrama de Atividade.....	38

Sumário

1.INTRODUÇÃO.....	8
1.1. Objetivos Gerais.....	8
1.2. Objetivos Específicos.....	8
1.3. Justificativa.....	8
2.FUNDAMENTAÇÃO TEÓRICA.....	9
2.1. NUI (Natural User Interface).....	9
2.2. Sensor Movimentos Kinect.....	11
2.3. Hardware do Kinect.....	12
2.4. Software Development Kit – Kinect.....	13
2.5. PowerPoint.....	14
2.6. InterKinectFace.....	15
2.6.1. Iniciar a apresentação em PowerPoint.....	17
2.6.2. Formas de interação com o Kinect.....	18
2.6.3. Dificuldades no desenvolvimento.....	25
3. VISÃO DO PROJETO.....	27
3.1. Introdução.....	27
3.2. Escopo.....	27
3.3. Processos Informatizados no Sistema.....	28
3.4. Principais Características:.....	28
3.5. Descrição do Problema.....	28
3.6. Definição da Posição do Produto.....	28
3.7. Principais Necessidades dos Usuários Envolvidos com o produto.....	29
3.8. Resumo dos Principais Recursos.....	29
3.9. Restrições.....	30
3.10. Riscos.....	30
3.11. Requisitos Mínimos do Sistema.....	30
3.11.1. Hardware.....	30
3.11.2. Software.....	30
3.12. Equipe do Projetos.....	31
3.13. Ambiente de Desenvolvimento.....	32
3.14. Cronograma.....	33
4- MODELAGEM.....	34
4.1. Diagrama de caso de uso.....	34

4.1.1. Geral.....	34
4.1.2. Por Assunto.....	35
4.2. Fluxo primário e secundário dos casos de Uso.....	35
4.3. Diagrama de Classes.....	36
4.4. Diagrama de Sequência.....	37
4.5. Diagrama de Atividades.....	38
.....	38
5-CONCLUSÃO.....	38
Referências.....	40

1. INTRODUÇÃO

1.1. Objetivos Gerais

Estudo e desenvolvimento de um projeto de software que possibilite a interação natural do usuário, através de gestos, movimentos ou voz utilizando o sensor de movimentos Kinect da Microsoft.

Nesta plataforma proporcionam-se os menus para interação, o controle do comportamento do mouse, e a criação ou adaptação dos mecanismos de interação para o conceito de NUI.

1.2. Objetivos Específicos

Estudar o hardware do sensor Kinect e a forma como ele funciona;

Estudar as *Application Programming Interface (API)*¹ de desenvolvimento próprias do kinect;

Estudar como o Windows interage com seus periféricos tais como mouse e teclado para que seja possível intervir no ponto exato desta interação simulando os comandos.

Estudar as práticas de desenvolvimento com interfaces NUI;

Utilização dos conceitos aprendidos, para criação de um projeto com foco em apresentações interativas, levando o conceito de NUI para aplicações do tipo PowerPoint e Prezi.

1.3. Justificativa

De frente ao desafio de criar uma plataforma NUI, foi realizada uma análise de softwares conhecidos, que tenham uma interação aceitável, mas que poderiam ser simplificados e melhorados adotando a utilização da interação NUI.

Apresentações de slides aos moldes do PowerPoint foi uma escolha bem compatível com o conceito.

¹ Interface de Programação de Aplicativos é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços. (<http://pt.wikipedia.org/wiki/API>) Acesso em 14/06/14

Em análise percebe-se que o ato de apresentar conteúdo através de slides em palestras ou eventos utilizando NUI é perfeito, pois o apresentador precisa apenas dos seus gestos e habilidades que traz consigo para mostrar o conteúdo, não dependendo mais de equipamentos como canetas laser, mouses ou amigos que realizem as transições dos slides, tornando o foco para o conteúdo da apresentação e não mais dispersando a concentração e atenção adquirida da plateia, enquanto muda de um slide para outro, criando uma cadência fluida e simples durante suas apresentações.

Este projeto pelo fato de conter elementos comuns a maioria das interações por NUI, permite o reaproveitamento em outros projetos, dado que o objetivo principal é uma interface genérica para criação de interfaces NUI, neste caso são utilizados os recursos como mapeamento de esqueleto, identificação de gestos, poses e controle do mouse, que poderá ser aplicado a qualquer outro contexto.

E com estes recursos posso apresentar as principais funcionalidades do sensor Kinect adotando os padrões de NUI.

2.FUNDAMENTAÇÃO TEÓRICA

2.1. NUI (*Natural User Interface*)

NUI é uma interface que faz com que seu usuário possa agir e sentir de uma forma natural. (Wigdor e Wixon 2011 no livro *Brave NUI World*).

NUI é considerada a próxima geração de interfaces, e uma nova forma de pensar sobre interação com dispositivos computacionais, podemos imaginá-la considerando a seguinte analogia (CLI → GUI → NUI) onde:

**CLI (Command Line Interface)*, interação através de comandos escritos e interpretados por um *prompt* de comando para desempenhar uma função específica em um aplicativo ou sistema operacional (Exemplos: MS-DOS, UNIX, LINUX);

**GUI (Graphical User Interface)*, janelas gráficas, mouses, botões e diversos outros componentes visuais que permitiam uma interação mais intuitiva e de fácil aprendizado, considerado um facilitador para a popularização dos computadores (Exemplos: Windows, Linux mais novos);

*NUI (*Natural User Interface*), interação é realizada com os recursos que são naturais ao utilizador.

Mas afinal de contas quais os conceitos de NUI? A proposito pronuncia-se “new-ee”.

Com a interface natural de usuário existem diferentes modalidades de entradas tais como multi toque, captura de gestos, movimentos e voz, para interagir com as aplicações diárias e o melhor, tudo que é preciso para isso existe naturalmente e já é utilizado para interagir com outras pessoas (seres humanos conversam, gesticulam, apontam etc..), por que não utilizar esta mesma forma para interagir com os dispositivos computacionais?

Uma boa definição para o termo NUI pode ser encontrada no livro “Natural User Interfaces in .NET” de Joshua Blake publicado pela Manning Publications <http://www.manning.com/>.

A NUI é uma interface de usuário desenhada para reutilizar as habilidades existentes para interagir direto com o conteúdo (Blake 2011, tradução nossa)².

Analizando GUI encontra-se elementos primários da interface janelas, menus e ícones transformando o conteúdo propriamente dito em elemento secundário (ocorre a interação com o conteúdo pelo intermédio destes componentes).

O foco do NUI é justamente a interação direto com o conteúdo, manipulando-o em primeiro plano, tendo como auxilio os componentes de tela como botões e *checkbox* quando necessário.

Como exemplo podemos citar a manipulação de uma imagem em um editor padrão de imagens, utilizando a GUI temos uma interface com diversos componentes visuais, temos um botão para aplicarmos zoom na imagem, ferramenta lápis para escrevermos, ferramenta para aplicar cores, neste cenário tem o foco primário nos componentes, que para uma boa interação precisamos conhecer suas funções o que exige certo tempo de aprendizagem, em NUI temos a imagem e utilizamos movimentos para aplicar o zoom, arrastamos as cores necessárias ao trechos desejados, neste abordagem NUI com o foco no conteúdo não precisamos de conhecimentos prévios dos componentes, precisamos apenas realizar movimentos intuitivos as ações desejadas, e é este o foco no conteúdo

2 A natural user interface is a user interface designed to reuse existings skill for interacting directly with content (blake 2011).

citado por Blake em sua definição, que é a proposta de NUI.

2.2. Sensor Movimentos Kinect

Proposta da Microsoft de interface natural de usuário (NUI), que vem a ser um hardware desenvolvido para atender os conceitos de NUI, trabalhando com um chip de reconhecimento desenvolvido pela empresa PrimeSense recém-adquirida pela Apple, como pode ser conferido nos endereços:

<http://www.theverge.com/2013/11/24/5141416/apple-confirms-primesense-acquisition>

<http://www.forbes.com/sites/shelisrael/2013/11/25/why-would-apple-buy-primesense/>

O Kinect possibilita uma interface natural ao usuário que ocorre interpretando os movimentos do corpo através de fluxos específicos de dados traduzindo-os em comandos para a aplicação, implementado inicialmente como substituto aos tradicionais controles do console Xbox360 da Microsoft em jogos especificamente desenvolvidos para o sensor e também para a interação com os menus do console.

Ocasinou uma nova revolução no mundo dos jogos proporcionando ao dispositivo o recorde mundial de venda mais rápida de um dispositivo de jogo, vendendo 50 milhões de unidades nos primeiros 60 dias de venda (133.333 unidades por dia), de acordo com as fontes a seguir:

<http://www.guinnessworldrecords.com/records-9000/fastest-selling-gaming-peripheral/>

<http://www.bbc.co.uk/news/business-12697975>)

sendo a chave deste sucesso a ideia de liberdade de controle, não dependendo mais de dispositivos como *joysticks*.

Mas o dispositivo até então estava limitado somente ao console e ao mundo do games, o próximo passo em 2012 foi o desenvolvimento de uma nova versão do sensor (Kinect para Windows), pensado especificamente para desenvolvimento de aplicativos da vida real.

2.3. Hardware do Kinect ³

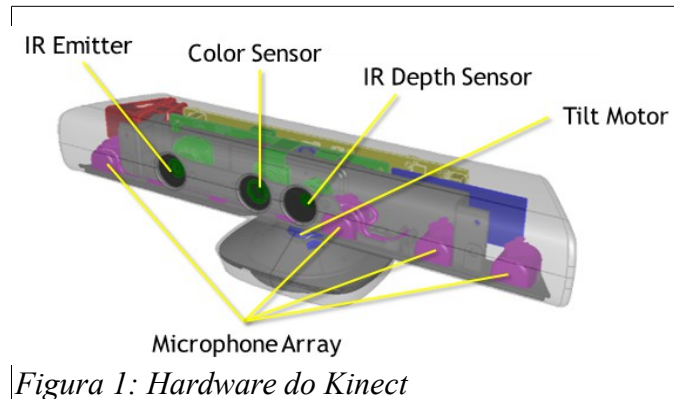


Figura 1: Hardware do Kinect

O Kinect é um dispositivo com sensores de profundidade, câmera de cor e um arranjo com microfones posicionados sobre o motor que controla a elevação do dispositivo.

A câmera de Cor é responsável pelo fluxo de dados de cor do vídeo, seu objetivo é detectar as cores vermelha, azul e verde, o fluxo que é retornado da câmera é uma sucessão de pequenos frames, mais precisamente 30 frames por segundo na resolução 640X480 pixels podendo chegar a 12 frames por segundo na máxima resolução de 1280x960 pixels.

A área de cobertura da câmera consiste em 43 graus na vertical por 57 graus na horizontal em posição fixa, considerando a movimentação do motor podemos mudar ângulos verticais em -27 até +27 graus.

³ *IR Emitter* (Emissor de infravermelho), *Color Sensor* (Sensor de cor câmera RGB), *IR Depth Sensor* (Sensor de profundidade infravermelho), *Tilt Motor* (Motor de Inclinação) *Microphone Array* (Arranjo de Microfones)

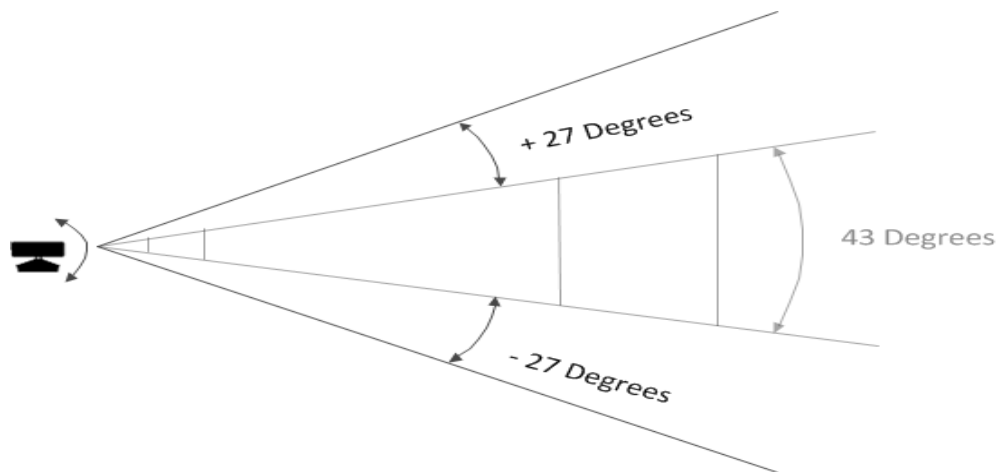


Figura 2: Kinect área de cobertura

O emissor de IR (infravermelho) e Sensor de Profundidade, trabalham juntos, o emissor de IR emite luz infravermelha em pontos invisíveis aos nossos olhos mas captados pelo sensor de profundidade, os pontos refletidos nos diferentes objetos são lidos pelo sensor e convertidos em informações sobre profundidade, medindo a distancia entre o sensor e os objetos diante deles, suporta as resoluções 640x480,320x240,80x60 *pixels*, o emissor e o sensor são gerenciados por um chip com a tecnologia da empresa PrimeSense.

O sensor kinect para Xbox tem a capacidade de rastrear objetos até 4 metros mas não consegue rastrear objetos muito perto abaixo de 80 cm, agora na versão para Windows ele é equipado com uma nova *firmware* que trabalha com o recurso *NearMode (Modo de perto)*, permitindo captura de objetos com distancia de 40 cm a frente do dispositivo sem perder a precisão no rastreamento.

2.4. Software Development Kit – Kinect

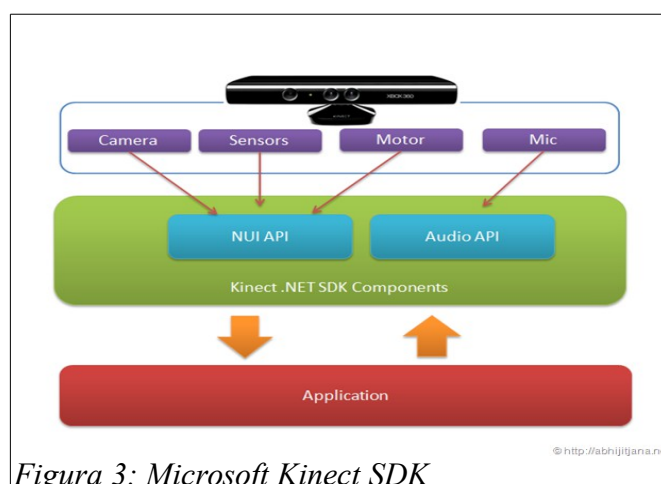


Figura 3: Microsoft Kinect SDK

Kinect SDK é o kit de desenvolvimento disponibilizado pela Microsoft “<http://www.microsoft.com/en-us/kinectforwindows/>(Microsoft, 2014)” que possibilita aos desenvolvedores criar aplicações NUI utilizando o sensor de movimento Kinect.

Atualmente esta na versão 1.8, devido ao lançamento de uma novo hardware o Kinect v2 estamos próximos do lançamento de uma nova versão da SDK.

Utilizando o Kinect SDK podemos desenvolver aplicações utilizando o Visual Studio e as linguagens de programação C#, C++, VB, tem também integrações com ambientes de desenvolvimento gráficos como XNA.

2.5. PowerPoint

Tudo começou em 14 agosto de 1984 com o desenvolvimento de um programa chamado Presenter da companhia Forethought Inc, pensado com o objetivo de criar slides de apresentações e foi lançado em 1987, neste mesmo ano a Microsoft comprou a empresa Forethought Inc por 28 milhões, ele foi aperfeiçoado pela equipe da Microsoft e um mês após a compra foi lançada uma nova versão com o nome de PowerPoint.

No começo o PowerPoint não era muito utilizado pelos profissionais por causa dos custos eles preferiam continuar usando suas transparências, na época somente grandes empresas utilizavam o PowerPoint.

Visando o aumento nas vendas a Microsoft realizou uma parceria com a empresa Genigraphics especializada em criar slides para grandes empresas e por isso possuía modelos padronizados de apresentações, desta forma criando os slides em cores para o PowerPoint, como fruto desta parceria foi lançada a versão 2.0 do PowerPoint.

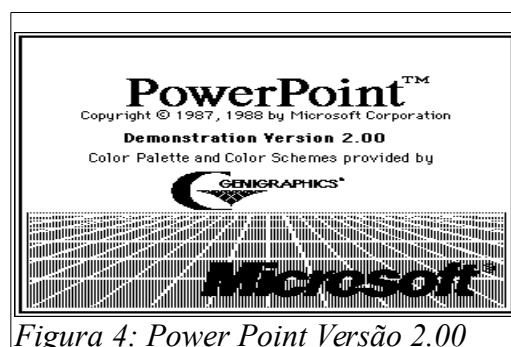


Figura 4: Power Point Versão 2.00

A Microsoft continuou a melhorar o PowerPoint e na década de 90 a Microsoft introduziu a Suite Office, reunindo seus produtos para escritório dentre os mais famosos o excel e o word, com a evolução do PowerPoint os consumidores começaram a perceber as facilidades de criar suas próprias apresentações resultando na perda do intermediário Geniraphics que criava os modelos de apresentações, libertando o PowerPoint que agora dependia apenas da criatividade de quem o utilizava, foi incluído recursos como vídeos imagens, hoje tornando praticamente o software padrão para criação e apresentação de slides.

Hoje o PowerPoint tem alguns concorrentes como o Prezi⁴ (<http://prezi.com/>) que tem o foco nas transições e estrutura diferenciada de tópicos e assuntos, mas mesmo assim o PowerPoint se mantém o software número um para apresentações de qualquer tipo, talvez pelas facilidades em criar apresentações, sejam corporativos, educacionais ou diversos fins, em sua lista de utilizadores temos o Steve Jobs que teve que utilizar o PowerPoint em algumas de suas apresentações pelo menos até a criação do keynote (<https://www.apple.com/br/mac/keynote/>) software de apresentações da Apple “<http://isachin.com/2010/10/28/apple-built-keynote-because-steve-jobs-hates/>(Agarwal, 2010)”

Discussões a parte, nascemos com a necessidade de expressar ideias, e o PowerPoint torna-se uma ferramenta poderosa pois muita vezes resumimos enormes quantidades de textos em apenas poucos slides.

Justamente pela popularidade do software em diversos ambientes de apresentação que optei por utilizá-lo, mas os recursos de interação utilizados por este projeto com o kinect, pelo fato de manipular os comandos nativos da máquina não está restrito apenas ao PowerPoint, pode ser utilizado em outros softwares de apresentação inclusive o Prezi necessitando apenas de pequenas alterações que fogem ao escopo deste trabalho.

2.6. InterKinectFace

Este projeto foi batizado com o nome de InterKinectFace, uma analogia as palavras “Interface e Kinect”, que juntas representam bem a ideia e o objetivo final.

4 O Prezi é um software na modalidade computação em nuvem feito em HTML5 utilizado para a criação de apresentações não lineares, e poderá substituir o comumente utilizado Power Point. No lugar, tudo é criado em uma estrutura única, parecida com uma palheta de designer real. (<http://pt.wikipedia.org/wiki/Prezi>) [Acesso](#) em 14/06/14.

Neste trabalho foi desenvolvido um sistema utilizando o sensor kinect como interface de interação com apresentações em PowerPoint, de forma que o apresentador não precise de nenhum tipo de equipamento que não o seus próprios movimentos para avançar e retroceder os slide.

Desta forma sempre que o apresentador movimentar a mão direita a uma determinada altura o programa interpretara este movimento traduzindo-o para instruções de comandos para a apresentação PowerPoint, possibilitando o avanço do slide atual, o mesmo ocorre quando o movimento é efetuado com a mão esquerda apenas muda-se a instrução ao PowerPoint que ao invés de avançar ira retroceder o slide.

A aplicação tem como plano futuro a implementação de comandos de voz, para que quando o apresentador utilizar os comandos de voz cadastrados, como exemplo ao dizer “*mouse*” o modo de interação poderá mudar de forma que o apresentador possa controlar o mouse utilizando suas mãos, a após basta dizer “*stop*” e o modelo de interação volta ao seu padrão inicial.

O Software pode ser utilizado para manipular apresentações em um notebook ou computador mas deve ser utilizado preferencialmente em ambientes que disponham de projetores e telas, tais como salas de reuniões, escolas e etc..

O software deverá possibilitar ao usuário configurar o diretório onde estão armazenados os arquivo com extensão PPT, para que só após seja possível selecionar a apresentação desejada, nesta primeira versão optei por não utilizar nenhum tipo de *login*, bastando para funcionar que seja executado, tendo como base para esta decisão o ambiente de utilização, que na maioria das vezes será em um auditório, áreas ou salas multimeios publicas ou seja o apresentador convidado chega e inicia sua apresentação descomplicada de criar login, acessar e etc..., futuras versões poderão ter o sistema de login mas no momento não é visto como essencial.

Apos configurar o diretório onde estão as aplicações o software realizará uma varredura neste diretório procurando por apresentações em PowerPoint criando um *thumbnail* com uma imagem em miniatura do primeiro slide de cada arquivo criando uma interface de seleção bem intuitiva, pois o apresentador reconhece visualmente sua apresentação, nesta tela já é utilizado as interações NUI bastando para

selecionar simular um clique movimentando a mão para frente enquanto o cursor estiver sobre a apresentação desejada.

Ao termino da apresentação o fluxo de execução é retornado para a tela de seleção de apresentação possibilitando o inicio de uma nova apresentação.

A ideia é basicamente utilizar o sensor de movimento Kinect para interpretar os movimentos de erguer a mão direita para avançar o slide ou erguer a mão esquerda para retroceder o *slide*, e como projeto futuro aperfeiçoar os controles de voz e a implementação de uma tela touch, transformando a tela projetada em uma lousa digital sensível ao toque.

2.6.1. Iniciar a apresentação em PowerPoint

Deve ocorrer após o clique no KinectTileButton que tem o evento tratado no método KinectTileButtonClick que realiza uma chamada ao método abrirPPT().

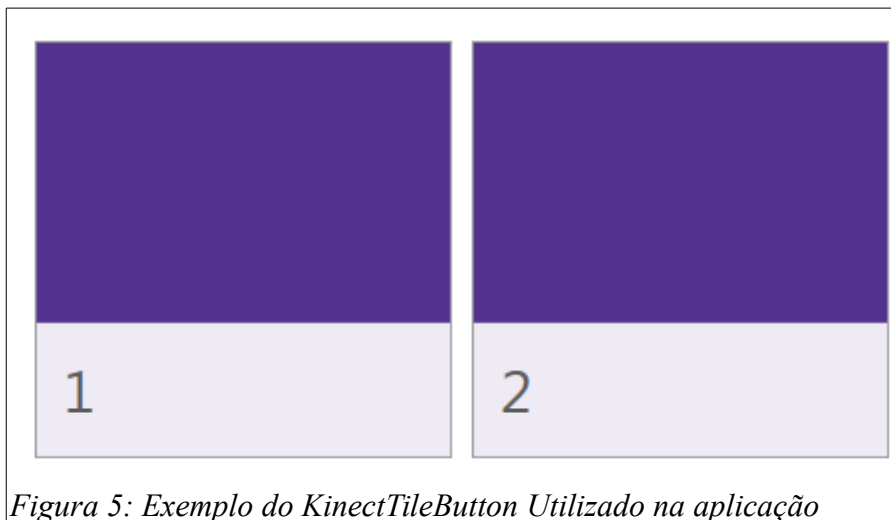
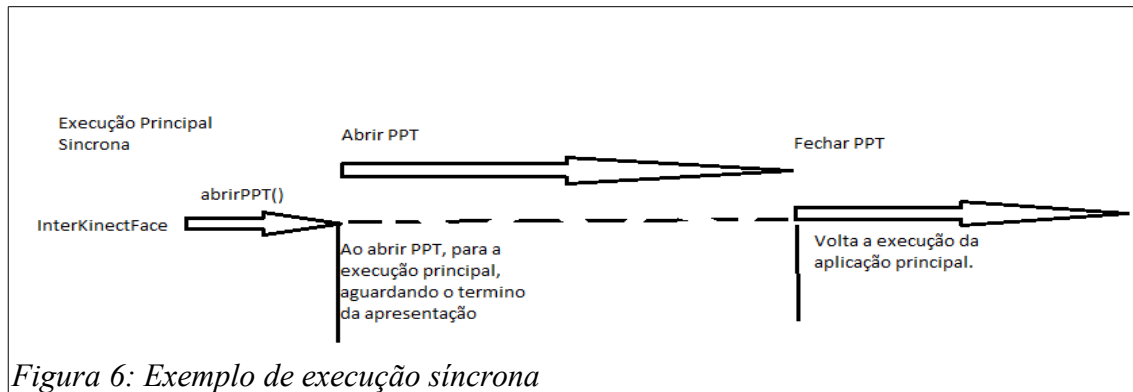
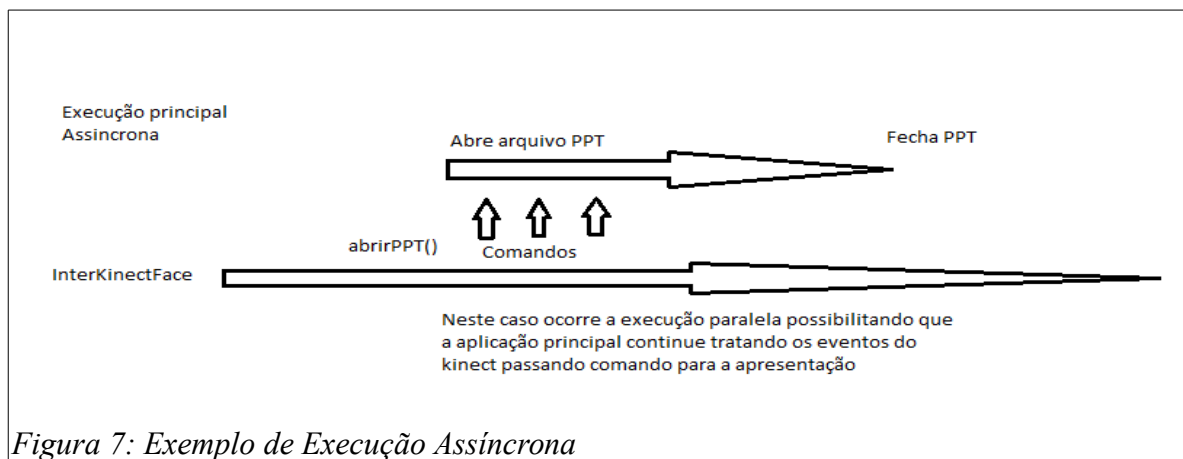


Figura 5: Exemplo do KinectTileButton Utilizado na aplicação

O método abrirPPT() inicializa a apresentação PowerPoint selecionada, porém como a apresentação inicializa-se a partir da aplicação corrente, enquanto a apresentação não terminar e retornar o fluxo de execução para a aplicação principal que fez a chamada não podemos tratar os fluxos do sensor Kinect. Isso ocorre pelo fato de a execução do programa utilizar métodos de chamada síncronos como mostra a figura a seguir:



Para melhorar esta execução utiliza-se o conceito de execução assíncrona, com o delegate que permite a chamada do método “abrirPPT()” que normalmente teria sua execução síncrona mas que pelo beginInvoke esta execução ocorre de forma assíncrona, com estes recursos mantemos a execução em paralelo da aplicação e da apresentação de PowerPoint possibilitando que os eventos do sensor Kinect sejam tratados e os comando enviados para a apresentação criando o ambiente de interação desejado. Como figura abaixo:



Neste projeto para realizar a manipulação dos arquivos PowerPoint utilizei as referencias dos componentes do namespace “Microsoft.Office.Interop.PowerPoint” (<http://msdn.microsoft.com/en-us/library/microsoft.office.interop.powerpoint.aspx>)

2.6.2. Formas de interação com o Kinect

Para que a interação com o Kinect seja possível é necessário obter os fluxos de dados disponibilizados pelo sensor Kinect, pois com estas informações é possível o tratamento na aplicação, para isso se utilizar os recursos e API's disponibilizadas

pela Microsoft para o Kinect:

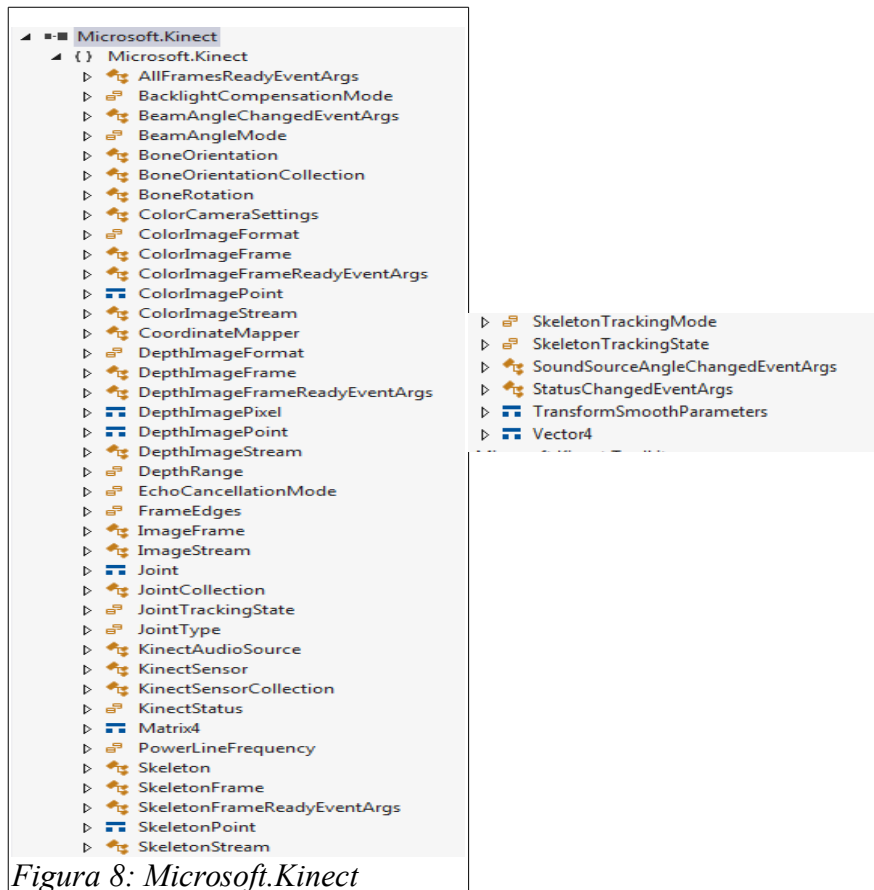


Figura 8: Microsoft.Kinect

Mesmo o Kinect SDK fornecendo os componentes necessários, ainda falta montar os processos de interação, principalmente telas e componentes visuais. Para isso temos o “Microsoft.Kinect.Toolkit” que traz recursos que simplificam o desenvolvimento tais como, a classe “KinectSensorChooser” que gerencia todo o ciclo de vida do sensor na aplicação, identificando o sensor kinect, preparando-o para utilização, tratando seus eventos de desconexão e erros, entregando para nossa aplicação o sensor funcional e pronto para utilização.

Neste *toolkit* existe também os componentes de controles do *Windows Presentation Foundation* (WPF) preparados para interação kinect, chamados de “KinectInteraction”, que nada mais são do que os componentes de tela próprios para interação com o Kinect.

No projeto foram utilizados os dois conceitos de interação, um utilizando todas

as facilidades do “*Microsoft.Kinect.Toolkit*” e seus componentes de tela para montar a tela de seleção do arquivos PowerPoint, e para interagir com a apresentação o método puro do “*Microsoft.Kinect*” tratando os fluxos de dados do Kinect para interagir com a apresentação em execução. Ambos explicados a seguir:

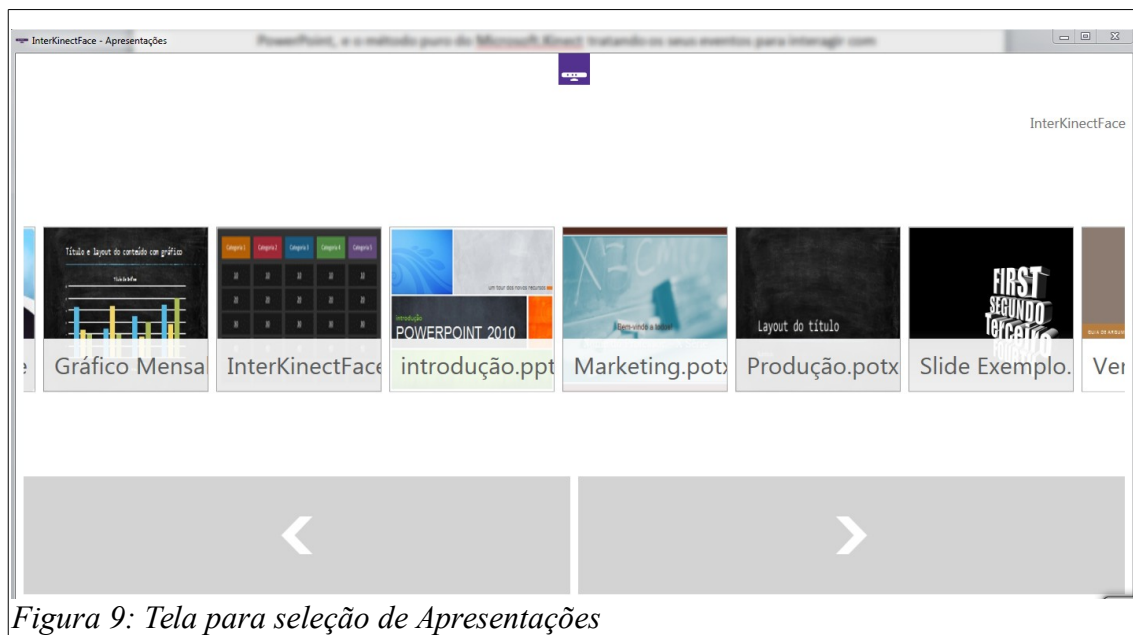
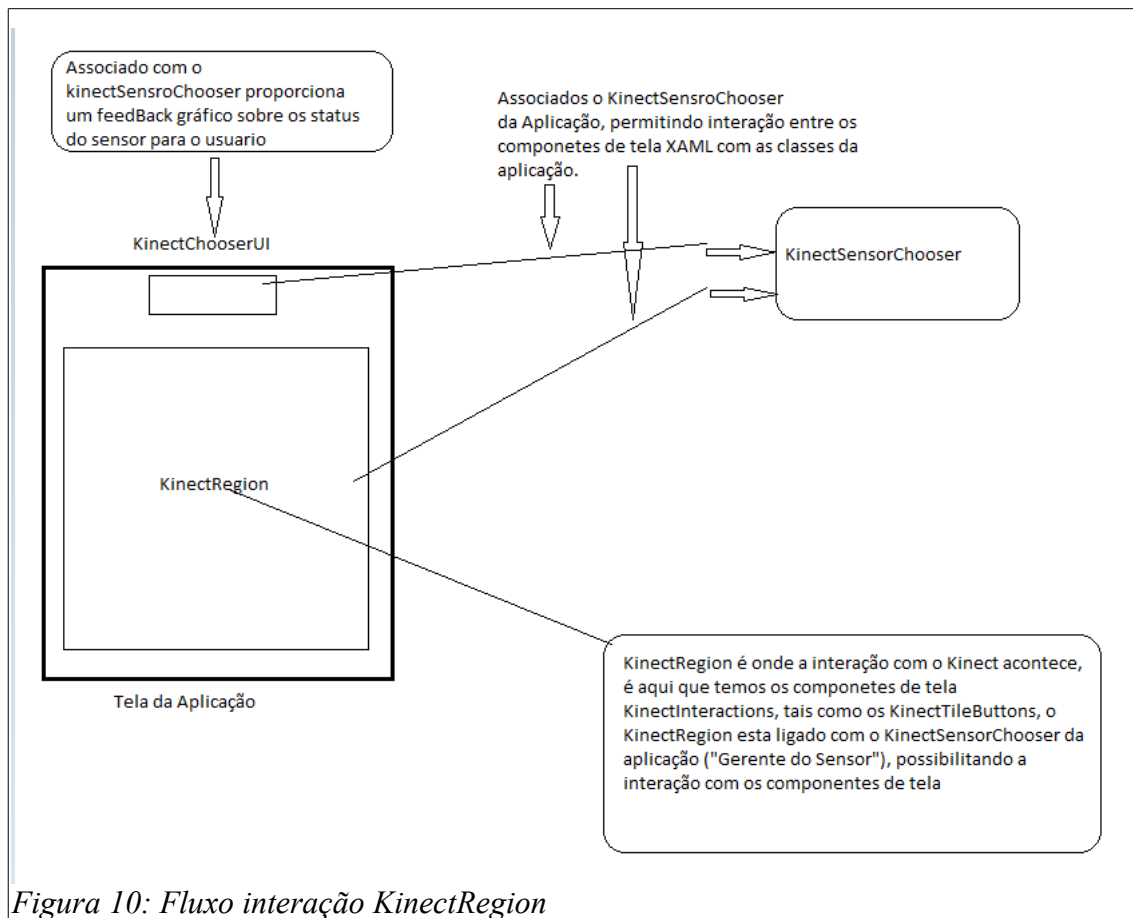


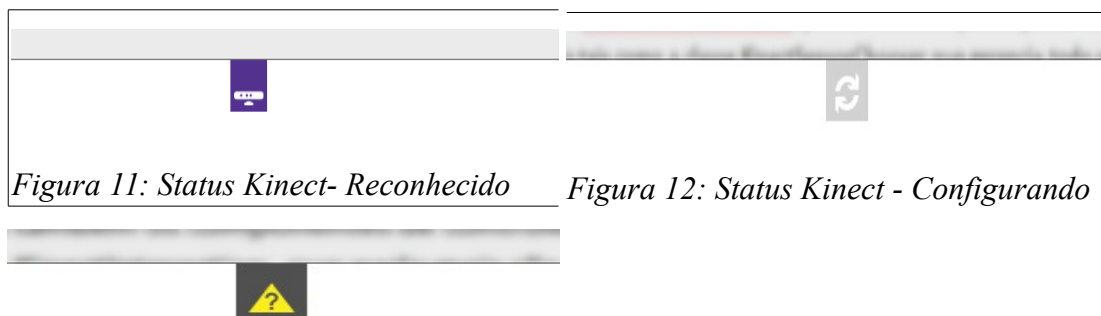
Figura 9: Tela para seleção de Apresentações

Interação utilizando os recursos do “*Microsoft.Kinect.Toolkit*” e “*KinectInteractions*”



Feito as associações e declarações exemplificadas na imagem acima o restante da interação fica por conta dos componentes.

Um componente interessante é o “KinectChooserUI”, que representa uma *feedback* gráfico para o utilizador informando se o sensor foi “reconhecido”, “sendo configurado” ou “não esta conectado”, demonstrado nas figuras a seguir:



Informando os diferentes status do sensor, é uma saída de tela para a classe

“KinectSensorChooser” pois ela gerencia o sensor passando as informações para o “KinectSensorUI” para que ele possa informar o utilizador.

A classe “KinectSensorChooser” garante também os devidos tratamentos no caso de desconexão repentina ou erros vindos por parte do sensor, prevenindo travamentos na aplicação.

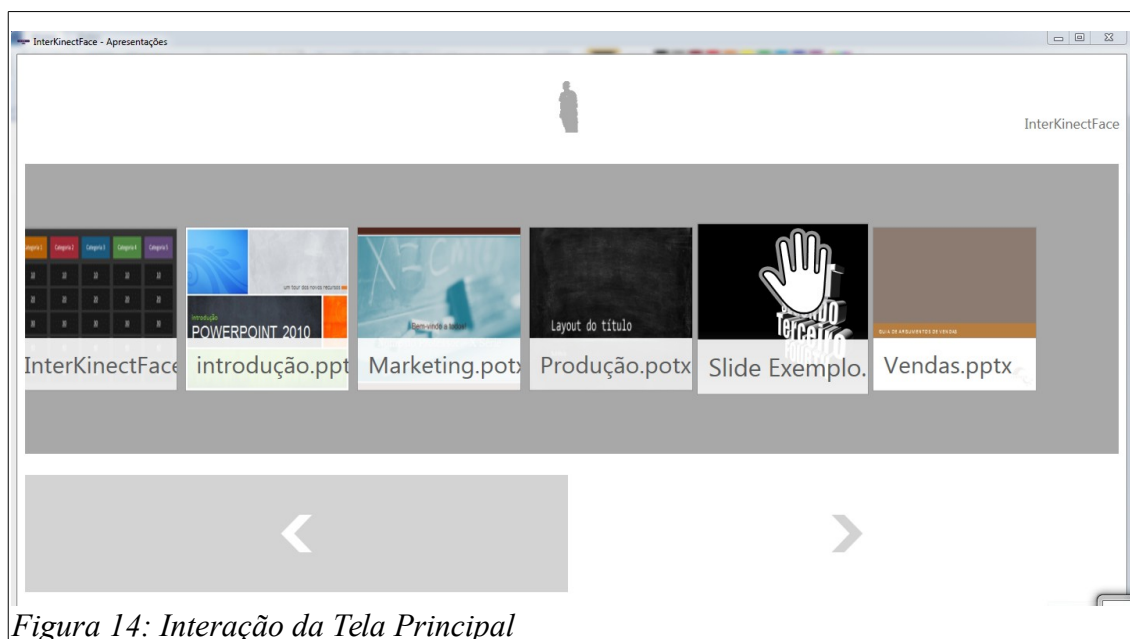


Figura 14: Interação da Tela Principal

Após seleção da apresentação de PowerPoint perde-se a possibilidade de utilizar estes recursos pois a apresentação não está em um “KinectRegion” mas sim em uma tela própria, neste caso utiliza-se as funções nativas para obter as informações do sensor preparando a interação.

O gerenciador do sensor “KinectSensorChooser” quando reconhece e habilita um sensor ele o disponibiliza através da propriedade “KinectSensorChooser.kinect”, basta registrar o evento nativo “SkeletonFrameReady” do sensor para um método da aplicação “sensor_skeletonFrameReady” para que sempre que disparado o evento o método seja executado.

O evento “SkeletonFrameReady” é o evento nativo do sensor disparado sempre que disponibiliza um *frame* com informações do esqueleto, no caso dispara 30 *frames* por segundo gerando um fluxo de dados sobre o esqueleto.

Em cada *frame* identifica-se um esqueleto válido ou reconhecido na aplicação referenciado pelo status de “*tracked*”, também se obtêm as posições no eixo X,Y,Z

de cada uma das 20 posições chamadas de juntas ou “*joints*”, um exemplo deste conceito é a manipulação do mouse, primeiro precisamos analisar no *frame* recebido a posição das mãos, o que pode ser feito da seguinte forma:

```
foreach (Skeleton sd in allSkeletons)
```

Para cada esqueleto declaramos as variáveis para manipular o curso do mouse e obtemos as informações das Duas mãos.

```
Joint jointRight = sd.Joints[JointType.HandRight];
```

```
Joint jointLeft = sd.Joints[JointType.HandLeft];
```

Obtemos as informações de posicionamento

```
cursorX = (int)scaledRight.Position.X;
```

```
cursorY = (int)scaledRight.Position.Y
```

Agora para manipular o *mouse* precisamos apenas sincronizar as coordenadas X,Y do ponteiro do *mouse*, utilizando a Dynamic-Link Library (DLL)⁵ que trata as entradas do mouse “user32” nativa do Windows com os da junta recém obtida, neste momento já estaríamos controlando o mouse .

```
[DllImport("user32.dll", SetLastError = true)]
```

```
private static extern uint SendInput(uint numInputs, Input[] inputs, int size);
```

A figura abaixo demonstra todas as 20 juntas tratadas pelo Kinect.

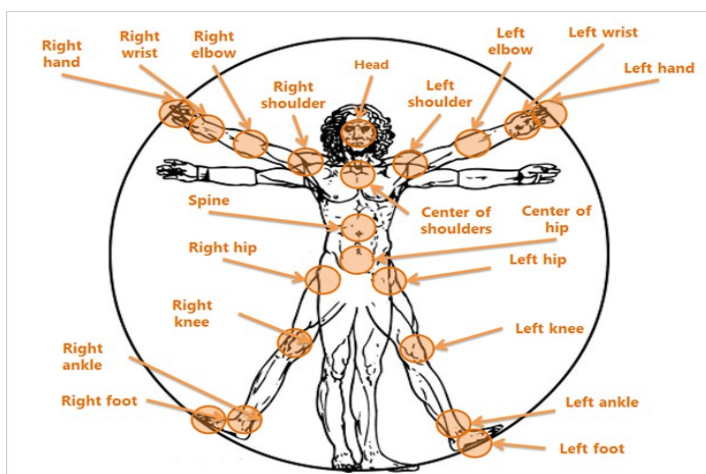


Figura 16: Kinect - Juntas do esqueleto

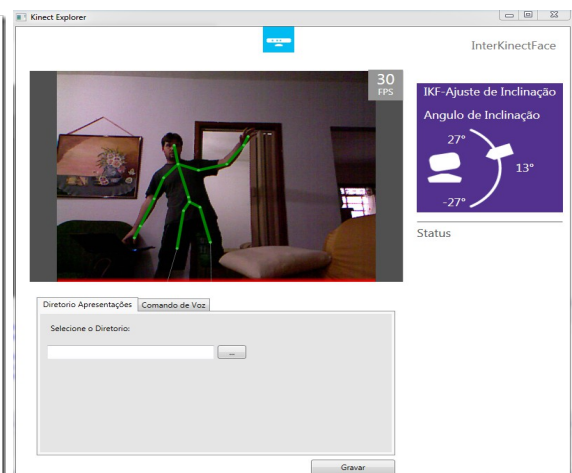


Figura 15: Juntas do esqueleto - IKF

5 Biblioteca de vínculo dinâmico ou DLL, é a implementação feita pela Microsoft para o conceito de bibliotecas compartilhadas nos sistemas operacionais Microsoft Windows e OS/2(<http://pt.wikipedia.org/wiki/DLL>) [Acesso](#) em 14/06/14.

Sendo *Right hand* (Mão Direita), *Right wrist* (Pulso Direito), *Right elbow* (Cotovelo Direito), *Right shoulder* (Ombro Direito), *Head* (Cabeça), *Right hip* (Quadril direito), *Right knee* (Joelho Direito), *Right ankle* (Tornozelo Direito), *Right foot* (Pé direito), *Spine* (espinha), *Center of hip* (Centro do quadril), *Center of shoulders* (Centro dos ombros), *Left shoulder* (Ombro esquerdo), *Left elbow* (Cotovelo esquerdo), *Left wrist* (Pulso esquerdo), *Left hand* (Mão esquerda), *Left hip* (Quadril esquerdo), *Left knee* (Joelho esquerdo), *Left ankle* (Tornozelo esquerdo), *Left foot* (Pé esquerdo).

Após obter as informações das mãos, utilizamos um método sobrecarregado pela referência “Coding4Fun.Kinect.WPF” disponível em (<http://c4fkinect.codeplex.com/>) chamado “ScaleTo”, pois ele possibilita que seja feito um ajuste entre as posições com relação ao monitor evitando diferenças desagradáveis entre a posição das mãos e do cursores do mouse como movimentar muito a mão e o cursor do mouse movimenta-se pouco.

Na figura abaixo temos o exemplo do fluxo completo de interação na aplicação de uma lado a interação utilizando os componentes de tela específicos e do outro tratando os eventos direto do fluxo de dados do sensor kinect.

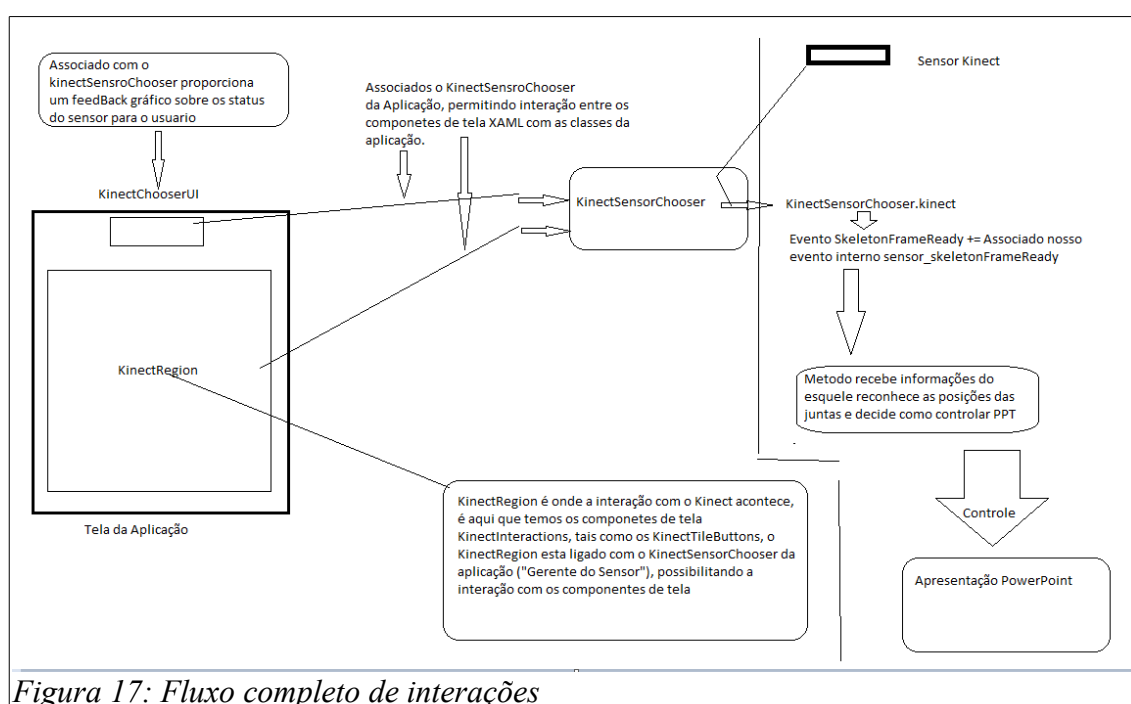


Figura 17: Fluxo completo de interações

As referências utilizadas “Microsoft.Kinect.Toolkit” e “KinectInteractions”, proporciona a abstração ao desenvolvedor dos eventos do Kinect, pois para montar a mesma tela aqui apresentada utilizando somente o Kinect SDK, ocasionaria em um trabalho extra, tratando os eventos, calculando posições das juntas e manipulando e sincronizando com os componentes de tela, então a ideia é que as interações mais genéricas de tela fiquem por conta das referências citada , mas a casos em que é necessário tratar os eventos vindos direto do sensor como neste projeto, pois é uma funcionalidade mais específica.

Até aqui recebe-se as informações do esqueleto, obtêm-se as posições das mãos, resta apenas enviar os comando adequado ao PowerPoint.

2.6.3. Dificuldades no desenvolvimento

Dentre as dificuldades encontradas merecem destaque as que proporcionaram uma maior satisfação e aprendizado ao superá-las.

Primeira foi a execução síncrona do software principal, que ao abrir a apresentação PowerPoint fica aguardando o termino da apresentação, ora como tratar os eventos do kinect e envia-los a apresentação se a execução da aplicação está parada esperando o término da apresentação?

Solução teórica seria utilizar conceitos de multiprocessamento, paralelismo e execuções assíncronas, inicia-se a pesquisa pelos *callbacks* do node.js ([“http://nodebr.com/callbacks-em-node/”](http://nodebr.com/callbacks-em-node/)) e *Compute Unified Device Architecture*, (CUDA) plataforma programação paralela para as GPUs da NVIDIA (http://www.nvidia.com.br/object/cuda_home_new_br.html), ao final conclui-se que a linguagem C# proporciona uma assistência muito boa a respeito de *MultiThred* e processos paralelos de execução, após entendidos estes conceitos aplica-se a solução de forma que a apresentação PowerPoint seja executada em paralelo com o aplicação principal possibilitando o envio dos comandos entre elas.

Segundo problema foi com relação ao que fazer após identificar o movimento desejado? como simular os eventos de entradas normais tais como pressionamento de teclas, e controles do *mouse*? outro ponto fundamental para a execução correta da aplicação.

Neste caso se inicia a pesquisa pelo mais óbvio, como o Windows controla seus periféricos de entrada, encontra-se as APIs próprias do Windows mais

precisamente o função “sendInput” da API “user32.dll” (<http://msdn.microsoft.com/en-us/library/Windows/desktop/ms646310%28v=vs.85%29.aspx>”), que contém os métodos utilizados pela maioria dos programas inclusive o sistema operacional para controlar estes dispositivos, por mais que a interação NUI seja uma nova tendência os métodos de baixo nível ainda são feitos pensando nos componentes de entrada padrão, a aplicação precisa simular os eventos destes componentes como se estivessem sendo pressionados, um exemplo é o controle do mouse na API user32.dll, existe um métodos que em seus parâmetros recebe as posições X e Y do mouse, considerando que com o Kinect podemos identificar as posições X e Y das mãos, basta apenas realizar chamadas aos métodos da API user32.dll com as posições em forma de parâmetros, desta forma simulamos o controle do mouse, o mesmo acontece com o teclado quando a mão do apresentador estiver a certa altura é simulado o pressionamento da tecla “seta para direita” o efeito na aplicação é o avanço ao próximo slide.

Como dito, na interação com aplicativos nativos ainda é necessário simular seus eventos nativos. Mesmo as interfaces totalmente criadas para o kinect com seu SDK kinect ainda existe a adaptabilidade citada, mas lembrando de que esta tecnologia de interface é nova está em processo de evolução.

Outra dificuldade enfrentada foi com relação ao testes de comando de voz que na versão utilizada suporta somente o Inglês, o que pode ser um problema ao apresentador que não tenha familiaridade com o idioma, foi realizado os testes tornando a função de comandos de voz funcional, porem o reconhecimento trabalha com um numero de avaliação da pronuncia, onde com base neste número o sensor decide se a palavra esperada foi pronunciado corretamente, deixando este valor muito baixo o sensor reconhece palavras similares mas não a esperada e executa o comando, neste caso não desejado, deixando o número muito alto ocorre o constrangimento do apresentador ao ter o seu comando ignorado pelo aplicativo, então deixo esta funcionalidade implementada apenas como exemplo do potencial do sensor e como possibilidade de melhora para próximas versões.

Outro ponto foi a familiaridade com o desenvolvimento utilizando a montagem de telas com *Extensible Application Markup Language*(XAML)/WPF, mais uma vez a pesquisa ajudou muito principalmente os códigos de exemplo que veem no próprio

SDK sem falar em sua documentação que é bastante intuitiva.

Outra dificuldade que merece ser comentada e a manipulação do PowerPoint pela aplicação como abrir a apresentação e inicia-la em modo de tela cheia foram encontradas diversas formas porem nenhuma se mostrou útil ao propósito, o que foi resolvido com o estudo dos recursos *interop*s da própria Microsoft.

3. VISÃO DO PROJETO

3.1. Introdução

Este documento tem como objetivo coletar e analisar os requisitos necessários ao desenvolvimento do software InterKinectFace, bem como levantar as necessidades e conhecimentos a serem adquiridos para completar o desenvolvimento do software em questão e satisfazer todos os requisitos definidos.

3.2. Escopo

O Sistema de apresentações interativas *“InterKinectFace”*, permite que sejam manipulados através de movimentos e gestos apresentações em PowerPoint (previamente criadas com o software da Microsoft), assim como a utilização opcional dos comandos de voz (em inglês) para auxiliar na interação.

O foco principal é a criação de uma interface com base em *NUI* (*“Natural User Interface” – interação homem maquina, que utiliza elementos naturais humanos como movimentos, gestos e voz*), possíveis de serem captados através dos sensores implementados em um dispositivo chamado Kinect, desenvolvido pela Microsoft em parceria com a empresa PrimeSense, que tem sua programação acessível através de uma biblioteca de desenvolvimento (Kinect SDK) disponibilizada gratuitamente pela Microsoft.

Neste projeto será utilizada a versão 1.7, do *SDK*, que disponibiliza todas as bibliotecas e recursos necessários para utilizarmos o Visual Studio 2012 e a linguagem de programação C# no desenvolvimento.

Todas as funcionalidades apresentadas estarão disponíveis através de um menu que primeiramente possibilitará configurações como, o diretório das apresentações PPT e utilização dos comando de voz, após a escolha do arquivo de

apresentação PPT e por fim a apresentação, onde toda a interação será realizada pelos gestos do apresentador.

3.3. Processos Informatizados no Sistema

- Processo de interação com apresentações PowerPoint.

3.4. Principais Características:

- O usuário poderá avançar e retroceder os slides de suas apresentações utilizando meios naturais para a interação, tais como gestos e poses;
- Ao iniciar as apresentações o ponteiro do mouse fica sob o controle da mão do usuário, permitindo durante a apresentação destacar e apontar trechos do slide em evidência;
- As apresentações tornam-se mais dinâmicas e interessantes;

3.5. Descrição do Problema

Descrição do Problema	
O(s) problema(s)	Poucas ferramentas existentes que proporcionam uma interface para interação com as apresentações, além da própria interface do software PowerPoint.
Afeta	Pessoas que gostariam de inovar em suas apresentações .
Cujo impacto é	Todas as apresentações acabam com a mesma dinâmica, sendo apenas mais uma apresentação de PowerPoint.
Uma boa solução seria	Aproveitar os recursos naturais disponíveis ao apresentador durante sua apresentação tais como seus gestos, movimentos e voz, também conhecido como NUI (<i>Natural User Interface</i>), que combinado com sensor de movimentos Kinect ,possibilita realizar todas as interações com sua apresentação, como avançar e retroceder os <i>slides</i> , realizar zoom na tela, controlar o ponteiro do mouse para destacar trechos no <i>slide</i> e tudo utilizando seus recursos naturais.

3.6. Definição da Posição do Produto

Definição da Posição do Produto	
Para	Professores, palestrantes, estudantes ou qualquer pessoa.
Que	precise realizar uma apresentação de forma inovadora e dinâmica.

O projeto	É um sistema de apresentações interativas através da NUI
Que	Visa proporcionar uma forma criativa e inovadora de apresentar aulas, palestras, reuniões corporativas etc...
Diferente de	Uma apresentação convencional onde apenas passamos <i>slides</i> com auxílio de um mouse sem fio ou algum colega que se disponha a fazê-lo, agora podemos interagir com nosso próprio corpo.
Nosso Produto/Projeto	Reúne as seguintes características: Apresentações Dinâmicas; Utilização da NUI (Natural User Interface); Facilidades na interação apresentador/apresentação.

3.7. Principais Necessidades dos Usuários Envolvidos com o produto

Principais Necessidades dos Usuários				
Necessidade	Prioridade	Preocupações	Solução Atual	Nosso Projeto
Forma dinâmica de interação com suas apresentações.	Alta	Criar apresentações inovadoras.	Projetor, Mouse sem fio e caneta laser	Permite interação utilizando linguagem gestual (NUI)

3.8. Resumo dos Principais Recursos

Principais Recursos do Sistema	
Recursos	Benefícios
Mapeamento do Esqueleto	Ponto principal para que tudo funcione, precisamos identificar e mapear o esqueleto do usuário.
Reconhecimento de Poses e Gestos	Com as poses e gestos associadas aos eventos corretos de interação podemos tratar toda a interface da apresentação em questão.
Interações com apresentações em PowerPoint utilizando NUI (Natural User Interface)	Torna a apresentação dinâmica, dispensando qualquer dispositivo tais como (mouse, transmissores, caneta laser etc...) simplificando tudo com gestos e movimento.

3.9. Restrições

De acordo com levantamentos e estudos sobre as possibilidades utilizando NUI (*Natural User Interface*), neste primeiro momento o software InterKinectFace não implementará o recurso de tela *touch*, que consiste em adicionar à tela de projeção uma função de tela *touch*, transformando-a por alguns momentos em uma lousa digital, recurso este que em primeiro momento é visto como melhoria para próximas versões. E não interfere na proposta principal do software que seria disponibilizar a NUI como interação para apresentações PowerPoint.

3.10. Riscos

Levantamento dos principais riscos identificados

Tipo	Descrição	Probabilidade	Impacto	Ações para evita-lo
Técnico	Falta de compatibilidade com outras versões do Kinect SDK	Baixa	Baixo	Distribuir instalador da versão adequada do SDK utilizado.

3.11. Requisitos Mínimos do Sistema

3.11.1. Hardware

- Computador ou Notebook
 - Processador de 32-bit (x86) ou 64-bit (x64);
 - Dual-core, 2.66-GHz ou maior capacidade;
 - Porta USB 2.0 dedicada ao Kinect;
 - 2 GB de RAM;
 - Placa de vídeo com suporte ao DirectX 9.0c;
 - Saída RGB ou HDMI compatível com projetor disponível;
- Sensor de movimento Kinect
 - Kinect para Xbox 360 ou Kinect para Windows.
- Projetor de acordo com as necessidades e compatibilidades de hardware

3.11.2. Software

- Windows 7 (32 ou 64 bits);
- .NET Framework 4;
- Kinect for Windows SDK 1.7 (March 2013);
- PowerPoint 2010 Home & Business ou maior;

3.12. Equipe do Projetos

Equipe do Projeto			
Papel	Nome	Atividade	Contato
Orientador do TCC	Carlos Alberto	Orientar os trabalhos de desenvolvimento do projeto	
Gestor do Projeto	Leonardo Bognesi	Gerenciar alocação de recursos para o desenvolvimento do projeto	leonardonhesi@gmail.com
Controle de Qualidade	Leonardo Bognesi	Auferir a qualidade do projeto	leonardonhesi@gmail.com
Analista de Requisitos	Leonardo Bognesi	Coletar e analisar os requisitos	leonardonhesi@gmail.com
Arquiteto	Leonardo Bognesi	Arquitetura a ser adotada no projeto	leonardonhesi@gmail.com
Projetista	Leonardo Bognesi	Desenvolvimento do Projeto	leonardonhesi@gmail.com
Desenvolvedor	Leonardo Bognesi	Desenvolver o produto	leonardonhesi@gmail.com
Gerente de Testes	Juliana F. Bognesi	Elaboração dos testes e do plano de testes	jullybognesi@gmail.com
Analista de Teste	Juliana F. Bognesi	Execução dos testes	jullybognesi@gmail.com

3.13. Ambiente de Desenvolvimento

As ferramentas utilizadas durante o desenvolvimento do projeto estão relacionadas na tabela a seguir:

Ferramentas de Desenvolvimento	
Tipo de aplicação	Descrição da Ferramenta
Ambiente Integrado de Desenvolvimento	MS- Visual Studio 2012
Kit de Desenvolvimento de Software SDK	Kinect SDK 1.7

3.14. Cronograma

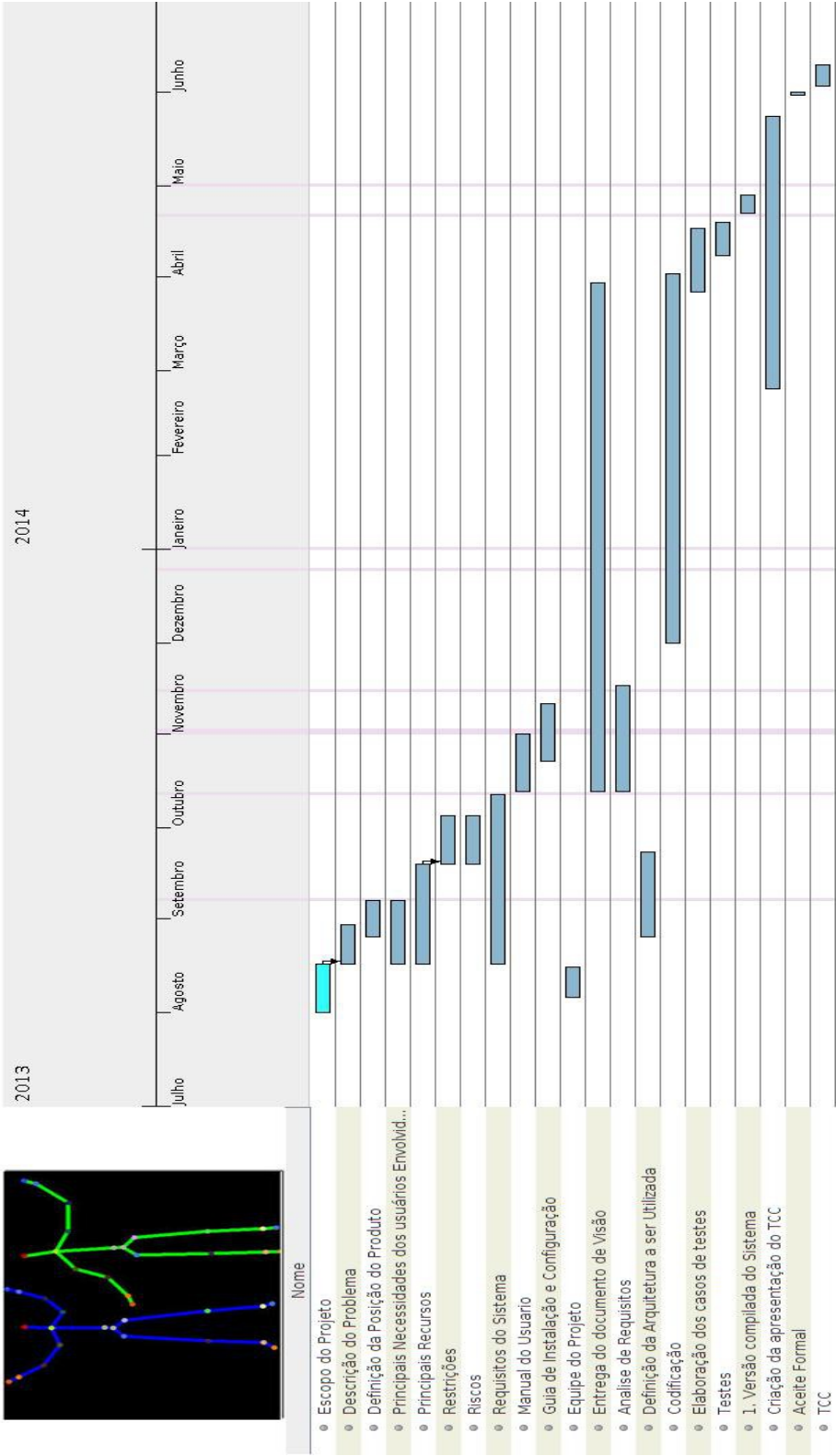


Figura 18: IKF- Cronograma do Projeto

4- MODELAGEM

4.1. Diagrama de caso de uso

4.1.1. Geral

IKF - Digrama Caso de Uso - Principal

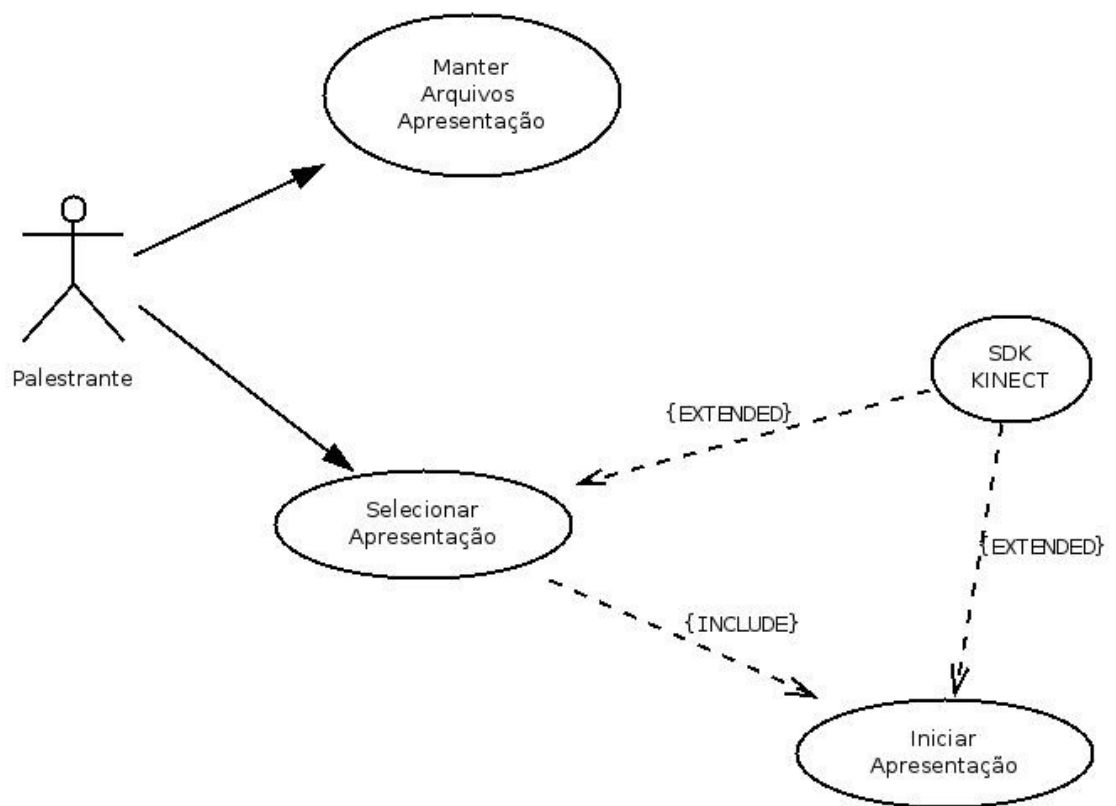


Figura 19: Diagrama Caso de Uso - Principal

4.1.2. Por Assunto

IKF-Diagrama caso de uso - Manter Arquivos de Apresentação

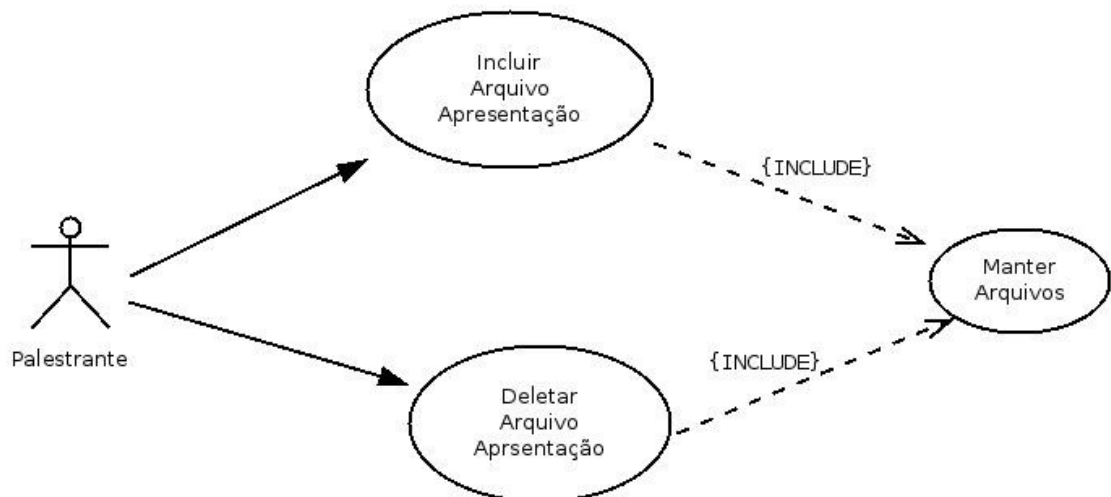


Figura 20: Diagrama caso de uso - Manter arquivos Apresentação

IKF-Diagrama Caso de uso - Iniciar Apresentação

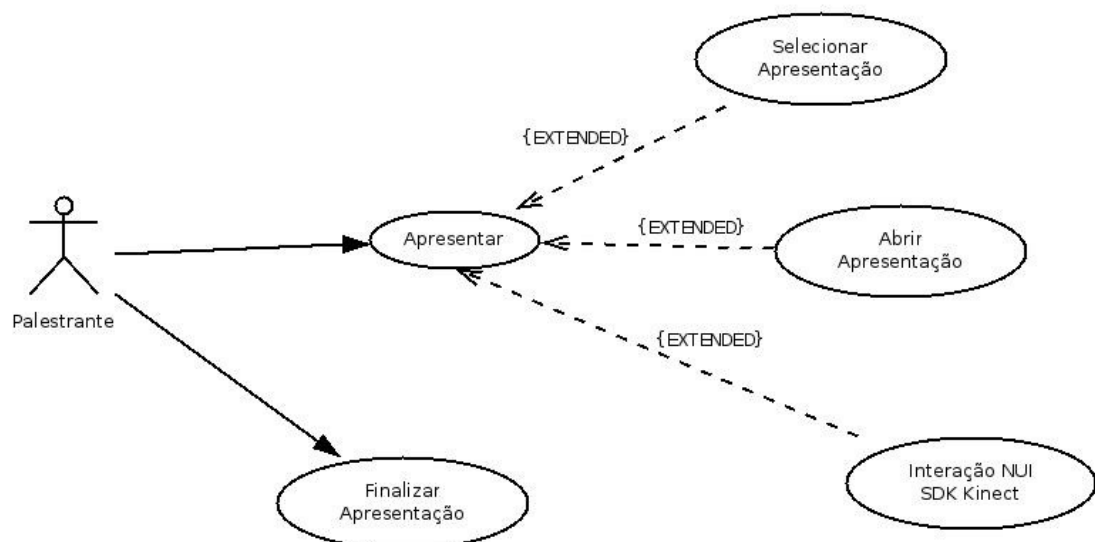


Figura 21: Diagrama caso de uso - Iniciar Apresentação

4.2. Fluxo primário e secundário dos casos de Uso

Fluxo primário principal (Inclui todos os casos de uso)

- 1-) Iniciar o software;
- 2-) Cadastrar arquivos apresentação;
- 3-) Selecionar Apresentação;

- 4-) Iniciar a apresentação;
- 5-) Interagir com a apresentação utilizando sensor Kinect;
- 6-) Finalizar apresentação;
- 7-) Retornar a tela de seleção de apresentação;
- 8-) Finalizar o Software;

Fluxo secundário

1-) No passo 1 ao iniciar o software deve-se verificar a existência de todos os recursos necessário (Software PowerPoint, Kinect SDK), informando o ocorrido impedindo a inicialização do software.

2-) No passo 2 ao cadastrar arquivos de apresentação deve-se verificar se os arquivos são realmente arquivos PowerPoint informando caso não seja.

3-) No passo 3 do fluxo primário caso não tenha apresentações cadastradas deve-se informar a ausência dos arquivos, oferecendo a tela para seleciona-los em disco.

4-) No passo 4 caso o sensor Kinect não esteja conectado e preparado deve-se avisar o utilizador do ocorrido e não inicializar a apresentação.

5-) No passo 5 caso o sensor seja desconectado durante a apresentação deve-se fecha-la, voltando para a tela inicial do software.

4.3. Diagrama de Classes

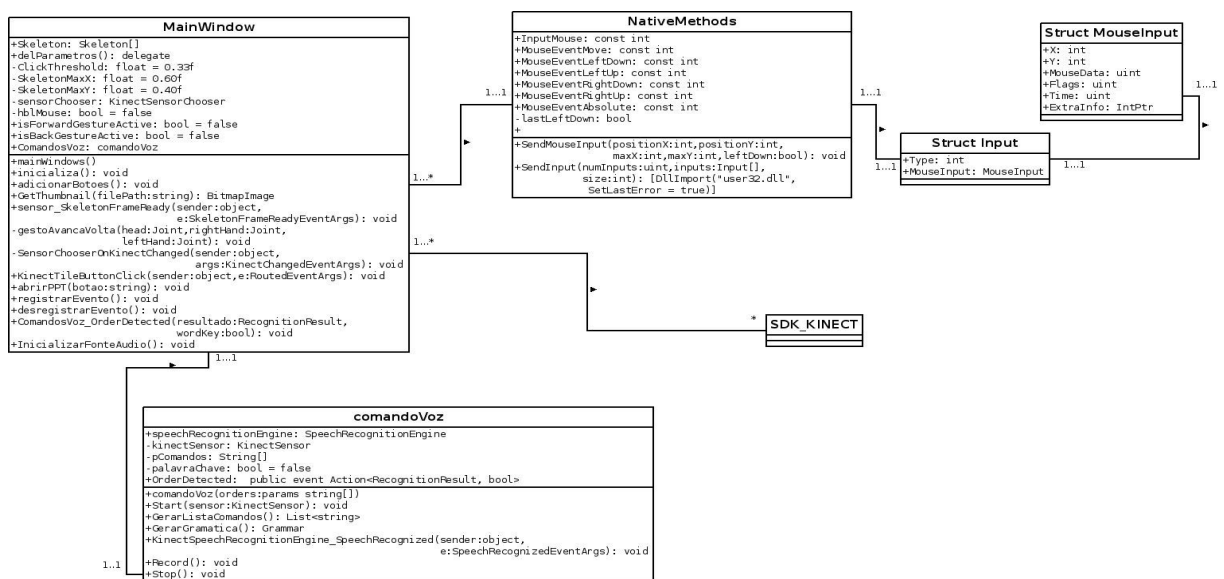


Figura 22: Diagrama de Classes

4.4. Diagrama de Sequência

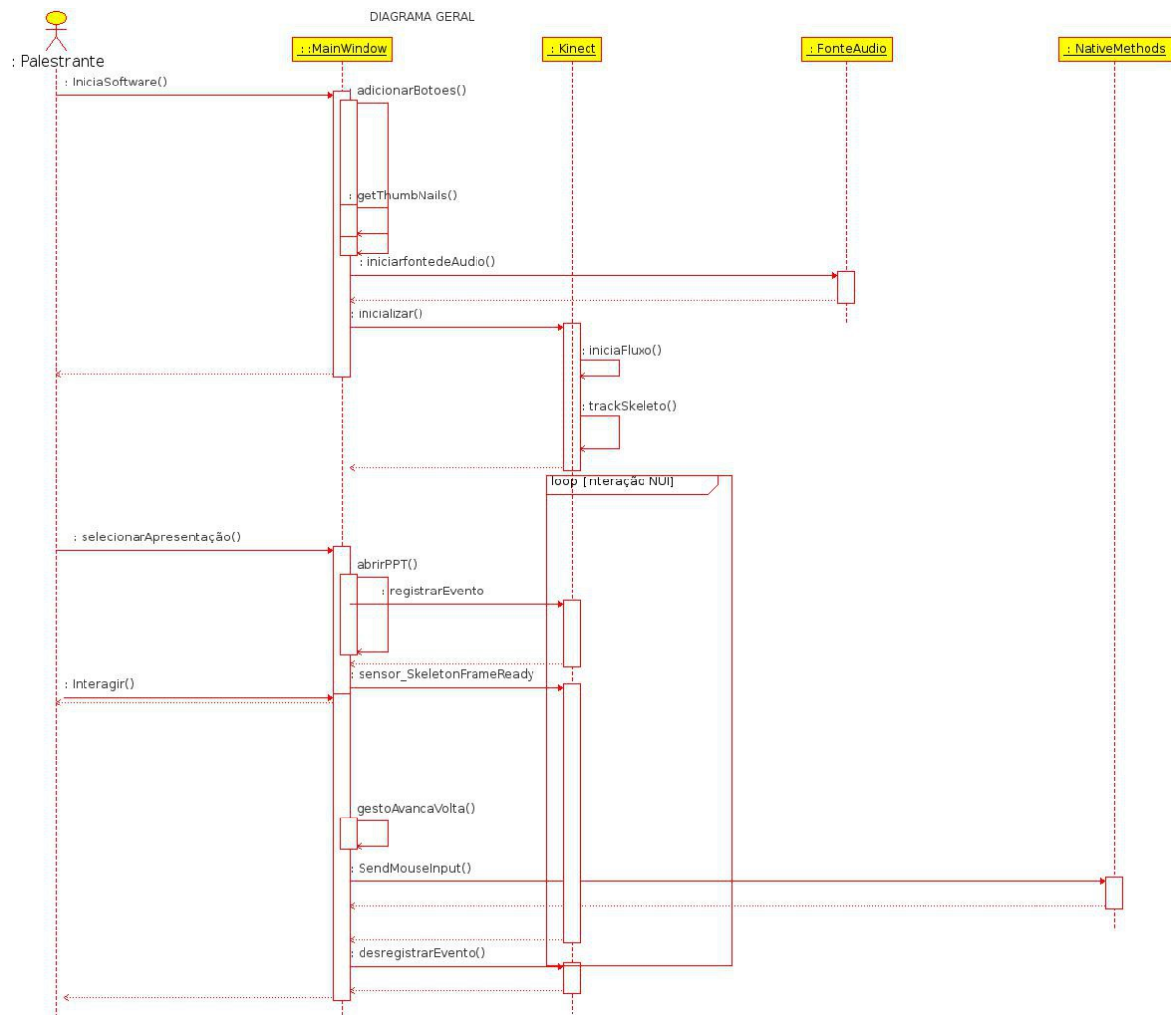


Figura 23: Diagrama de Sequência

4.5. Diagrama de Atividades

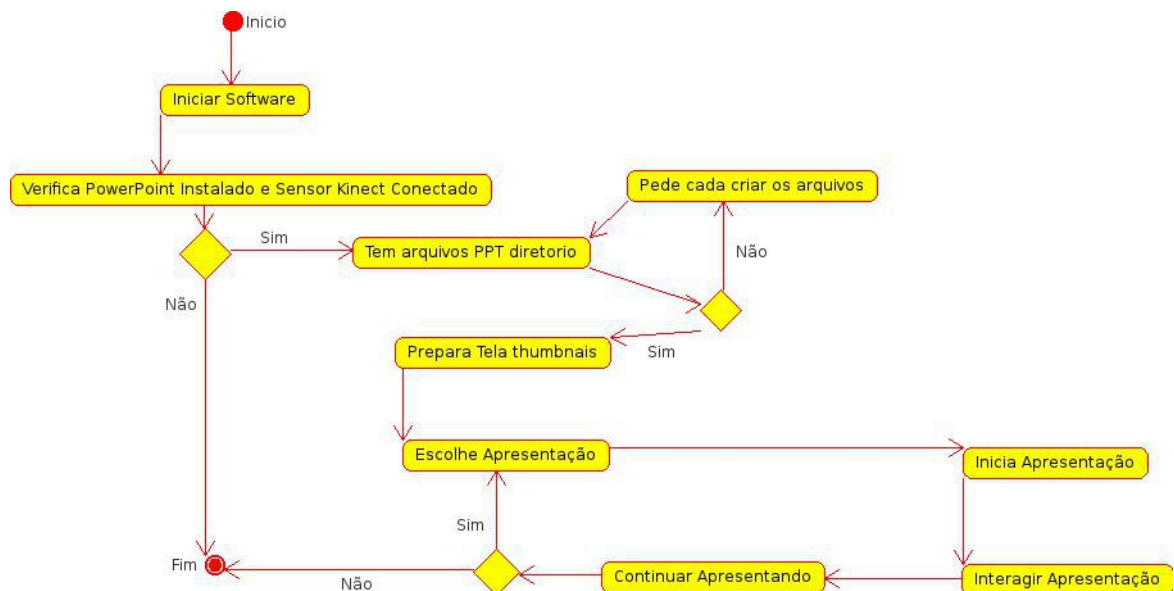


Figura 24: Diagrama de Atividade

5-CONCLUSÃO

A escolha do tema aqui tratado, proporcionou um desafio e a oportunidade de testar habilidades de pesquisa e aprendizado em uma aplicação totalmente nova.

Desde o início apresentaram-se distâncias razoáveis entre o que gostaria de fazer e o que a tecnologia poderia proporcionar, como todo projeto seu início foi pelo levantamento mas não ainda de requisitos, mas sim de possibilidades, respondendo perguntas tais como:

Com a disponibilização do SDK o que outros desenvolvedores estão fazendo ?

Quais os segmentos de mercado que estão interessados em pesquisas ou projetos com NUI ?

Neste período foram encontrados diversos projetos, dentre os quais se destaca o projeto do FIT - Instituto de Tecnologia, de Sorocaba que em 2012 criou um painel interativo explorando a questão da deficiência física como fator de inclusão, sendo o mais recente dos projetos da FIT um *quiz* com temas educativos controlados com movimentos (<https://www.youtube.com/watch?v=ihE6dpfsRcM>), também foram encontrados outros SDKs anteriores ao oficial da Microsoft, afinal

desde que foi lançado desenvolvedores buscam formas de utilizar o sensor em suas aplicações.

Ao ponto de deter base teórica satisfatória para o início resta apenas aplicá-las em uma solução real, desta forma então trazer os conceitos de NUI para as apresentações de PowerPoint.

Um fator importante neste tipo de programação é a criatividade e o domínio das informações fornecidas pelo sensor Kinect. Informações de Cor, Informações do Esqueleto e de profundidade.

Um pouco de matemática também foi fundamental afinal em alguns momentos é necessário trabalhar com as posições dos eixos X, Y e Z pois o sensor nos entrega informações tridimensionais.

A modelagem deste trabalho foi simples pois o foco não é uma aplicação empresarial a exemplo de *Enterprise Resourcing Planning* (ERP)⁶ com regras complexas de negócios, mas sim voltada ao desenvolvimento de hardware, software e estudo de interações humano máquina, mas como todo projeto embora simplificado foi utilizado os conceitos de modelagem adquiridos durante o período acadêmico, neste primeiro momento não sustenta-se a necessidade de utilizar banco de dados, pois a massa de dados processada pelo sistema do projeto tem como fonte o fluxo de dados obtidos em tempo de execução vindos direto do sensor Kinect.

Foi aperfeiçoado também conhecimentos de programação propriamente dita, principalmente os recursos da linguagem C#.

Com as possibilidades apresentadas ao decorrer do desenvolvimento foram iniciados alguns outros projetos paralelos utilizando o sensor Kinect + NUI, algumas integrações com o Arduino, um sistema de tela touch em superfícies projetadas, até mesmo um projeto para ensinar inglês dado que o reconhecimento de voz do Kinect pontua a pronúncia para decidir o que foi dito é a palavra esperada. Afinal este tipo de programação cativa o desenvolvedor, e pretendo continuar a explorar as possibilidades principalmente agora com a versão 2 do hardware que veio junto com a nova geração de consoles e sua nova SDK.

Assim considera-se que o objetivo principal foi atingido, afinal o software

6 ERP sistemas de informação que integram todos os dados e processos de uma organização em um único sistema.

proporciona as funcionalidades projetadas, pelo fato de ser uma tecnologia relativamente nova e em constante evolução deixa espaço para novas ideias afinal vivemos em um mundo de melhora continua.

Referências

ABHIJIT, Jana .**Kinect for Windows SDK Programming Guide**. Packt Publishing 2012.

BLAKE, Joshua. **Natural User Interfaces in .Net**. Manning 2011.

Cardoso , Gabriel Schade. **Microsoft Kinect: Crie aplicações interativas**. Casa do Código 2013

Chamando métodos síncronos assincronamente. Disponível em:

<<http://msdn.microsoft.com/pt-br/library/vstudio/2e08f6%28v=vs.100%29.aspx>>. Acesso em: 27 mai 2014.

_____.**Coding4Fun Kinect toolkit**. Disponível em:<<http://c4fkinect.codeplex.com/discussions>>. Acesso em: 27 mai 2014.

_____.**Delegate Métodos Assíncronos. 2013**. Disponível em:<<http://msdn.microsoft.com/pt-br/library/2e08f6yc%28v=vs.110%29.aspx>>. Acesso em:25 mai. 2014.

_____.**Introdução ao WPF** Disponível em:<<http://msdn.microsoft.com/pt-br/library/aa970268%28v=vs.110%29.aspx>>. Acesso em 25 mai 2014

_____.**Kinect for Windows SDK. 2014**. Disponível em:<<http://msdn.microsoft.com/en-us/library/hh855347.aspx>>. Acesso em: 25 mai. 2014.

Kinect for Windows SDK: Skeletal Tracking. 2013. Disponível em:

<<http://msdn.microsoft.com/en-us/library/hh973074>>. Acesso em: 23 abr. 2013.

Minority Report - A Nova Lei. Direção de Steven Spielberg .20th Century Fox e DreamWorks SKG Estados Unidos, 2002 145 min., Cor, DVD > Ficção Científica.

NUI, Group Authors. Multi-Touch Technologies . NUI Group 2009.

Painel Interativo para deficientes físicos FIT. Disponível em:<<http://g1.globo.com/tecnologia/noticia/2012/08/brasileiros-criam-painel-interativo-para-deficientes-fisicos-com-o-kinect.html>>. Acesso em: 25 mai 2014.

PowerPoint 2010 Primary Interop Assembly Reference. 2014. Disponível em:<<http://msdn.microsoft.com/en-us/library/office/ff759900%28v=office.14%29.aspx>>. Acesso em: 25 mai. 2014.

_____TedTalks -2014. Disponível em:<http://www.ted.com/talks/john_underkoffler_drive_3d_data_with_a_gesture>. Acesso em: 25 mai. 2014.

Visão Geral de XAML. Disponível em:<<http://msdn.microsoft.com/pt-br/library/ms752059%28v=vs.110%29.aspx>> Acesso em: 25 mai 2014.

WIGDOR, Daniel; WIXON, Dennis. **Brave NUI World: Designing Natural Interfaces for Touch and Gestures.** Burlington: Elsevier, 2011.

Wikipédia – Natural User Interface. 2013. Disponível em:<http://en.wikipedia.org/wiki/Natural_user_interface>. Acesso em 23 abr. 2013.