

# Kinect SDK ver. 1.8 sample report

神奈川工科大学情報学部情報メディア学科  
Team KR

0923100 棚橋陽太

1123032 鈴木久貴

1123050 小野禎樹

1123131 殿岡竣太

1123199 原田裕貴

# Kinect SDK ver. 1.8 samples

- 全65サンプルが存在
- 全サンプルの内、多言語重複が18件
- 全サンプルの内、起動オプションがないのが14件
- 本レポートでは25件を紹介する

## 担当

- 棚橋 サンプル実行担当
- 鈴木 英語読み、記録担当
- 殿岡 英語読み、記録担当
- 原田 サンプル実行担当、記録担当
- 小野 アクター

## 発表順①

Developer Toolkit v1.8.0 の上から順

- Adaptive UI-WPF
- Audio Basics-WPF
- Audio Capture Raw-Condole
- Audio Explorer-D2D
- Avateering
- Background Removal Basics-WPF
- Color Basic-WPF
- Coordinate Mapping Basics-WPF
- Controls Basic-WPF

## 発表順②

- Depth Bascis-WPF
- Depth-D3D
- Depth with Color-D3D
- Face Tracking 3D-WPF
- Face Tracking Basics-WPF
- Face Tracking Visualization
- Infrared Basics-WPF
- Interaction Gallery-WPF
- Kinect Explorer-WPF
- Shape Game

## 発表順③

- Skeleton Basics-WPF
- Speech Basics-WPF
- Tic Tac Toe-WPF
- WPF D3D Interop
- XNA Basics

# Adaptive UI-WPF

- Kinectが動作するための環境設定の実行と確認を行うプログラム。

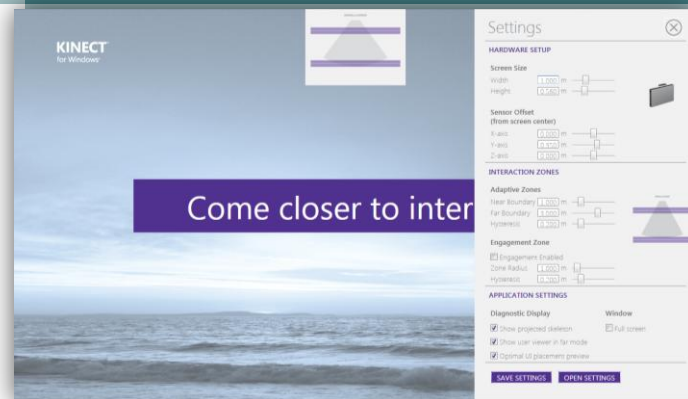


図1：起動画面

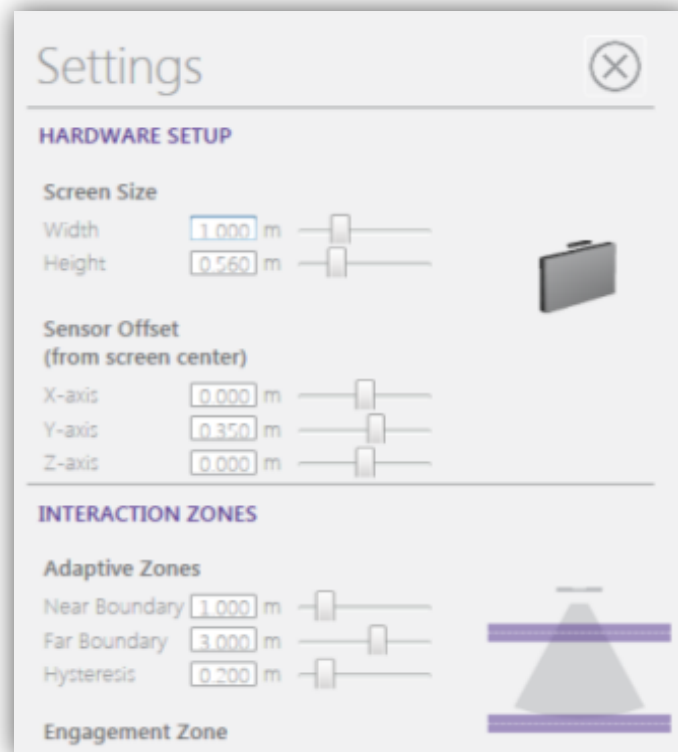


図2：設定メニュー

# Audio Basics-WPF

- Kinectが感知した音の「最新の方向」と「方向の信頼度」と「オーディオ信号強度」を表示するプログラム
- オーディオ信号強度をスクロール波形でリアルタイム可視化。
- ノイズの多い環境でも、常に大きな音がなっている方向にセンサが反応し、矢印で指し示す。
- WPF、D2D、WPF-VB版あり

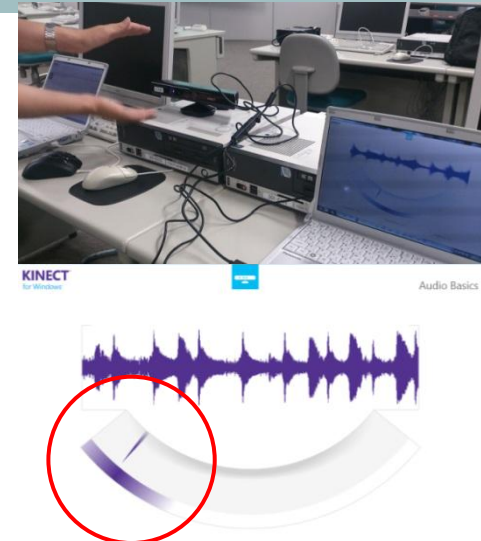


図3：左側で手を叩いた様子



図4：右側で手を叩いた様子

# Audio Capture Raw- Condole

- オーディオをキャプチャ(録音)するプログラム。ファイル形式はwaveファイル。
- Sキーを押すと、録音をストップできる。
- 録音したファイルの保存先は  
C:\Users\user\_name\Music\  
KinectAudio-HH-MM-SS.wav

※HH、MM、SSはそれぞれ  
保存された時間、分、秒。

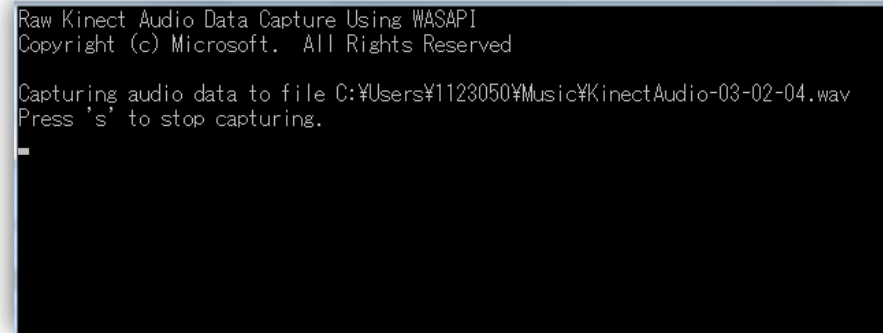


図5：起動画面



図6：録音している様子



# Audio Explorer-D2D

- オーディオの検出、録音を同時に行えるプログラム。
- オーディオの波形は、ビジュアルライザで表示。
- 矢印は、最新の音源の検出方向を示している。
- インジケータバーは、音源検出の信頼水準を示している。  
バーの長さは信頼度が高いほど色が濃くなり、信頼度が低いと色は薄くなる。



図7：手を叩いている様子

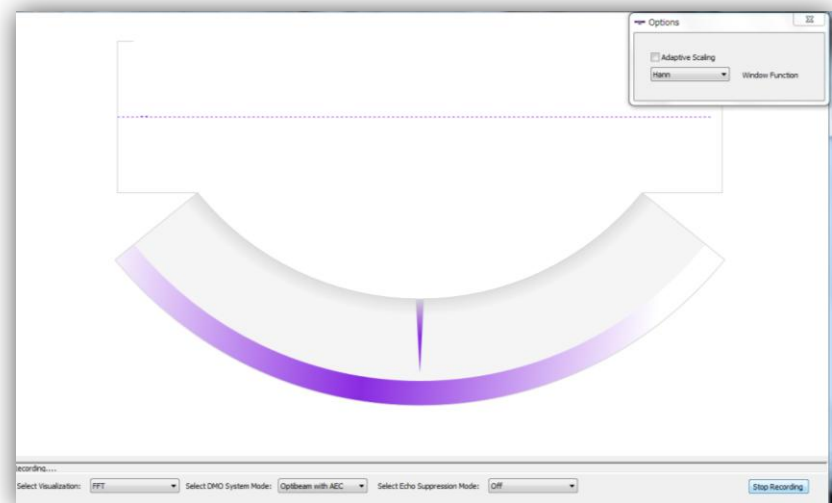


図8：起動画面

# Avateering

- Kinectが捉えた人の動きに合わせて アバターの男性が動作するプログラム。
- キーによりカメラを制御出来る。

左カーソル・・・左にパン

右カーソル・・・右にパン

Wキーまたは上カーソル・・・ティルトアップ

Sキーまたは下カーソル・・・ティルトダウン

Zキー・・・ズームアウト

Xキー・・・ズームイン

- Visual Studio2010でサンプルを開く場合、XNA Game Studio 4がインストールされている必要がある。



図9：ポーズをとっている様子



図10：起動画面

# Background Removal Basics

- 背景画像を撮影対象のスペースのみ切り抜いて合成するプログラム。スクリーンショットも可能。
- WPF,D2Dのうち、WPFの動作を確認した。

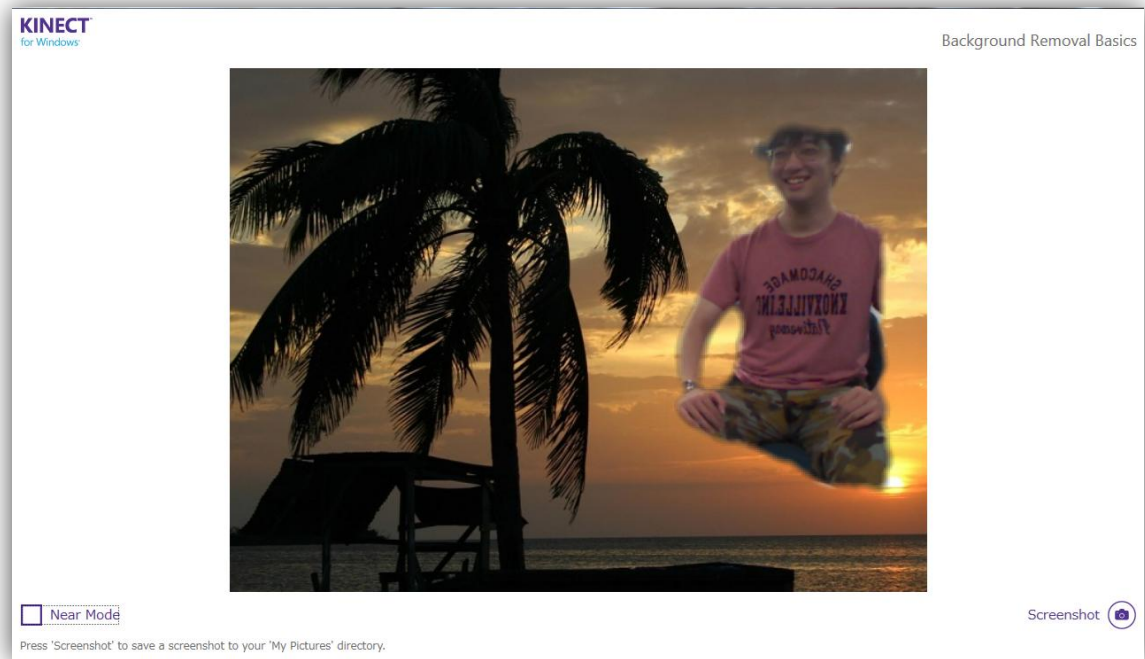


図11：撮影対象を背景と合成した状態

# Color Basics

- カメラで撮影してる映像をカラーで静止画として保存できる。  
ファイル形式はpng。
- WPF,D2D,WPF-VBのうち、WPFの動作を確認した。

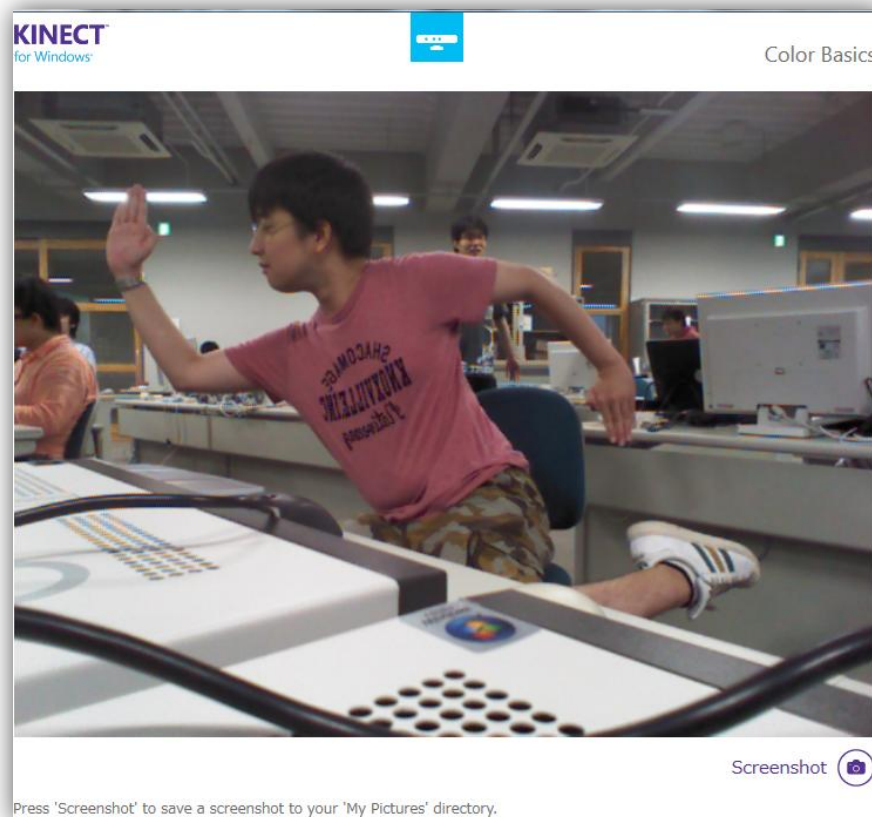


図12：カラー撮影によるスクリーンショット

# Coordinate Mapping Basics

- 背景静止画に撮影対象を重ねて合成描画する。スクリーンショットも可能。
- WPF,D2Dのうち、WPFの動作を確認した。

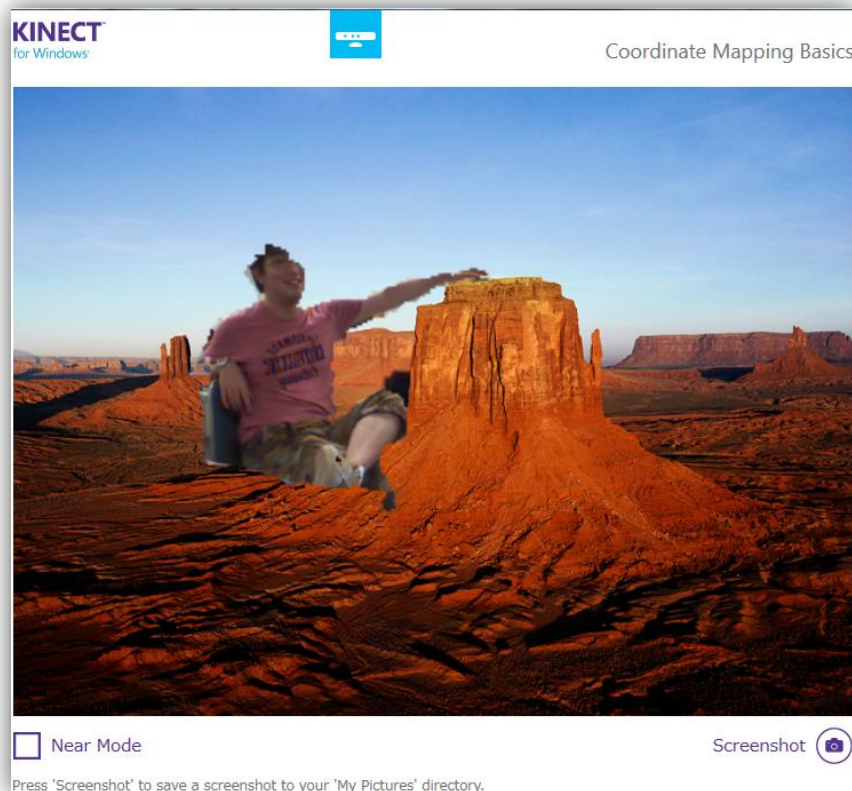


図13：背景画像上に対象を合成



# Controls Basics

- ユーザービューウィンドウが動作するプログラム。動作方法は「手を左右にスライド」「つかむ」「手を全面に押す」「手をそえる」である。右手、左手の認識も可能。

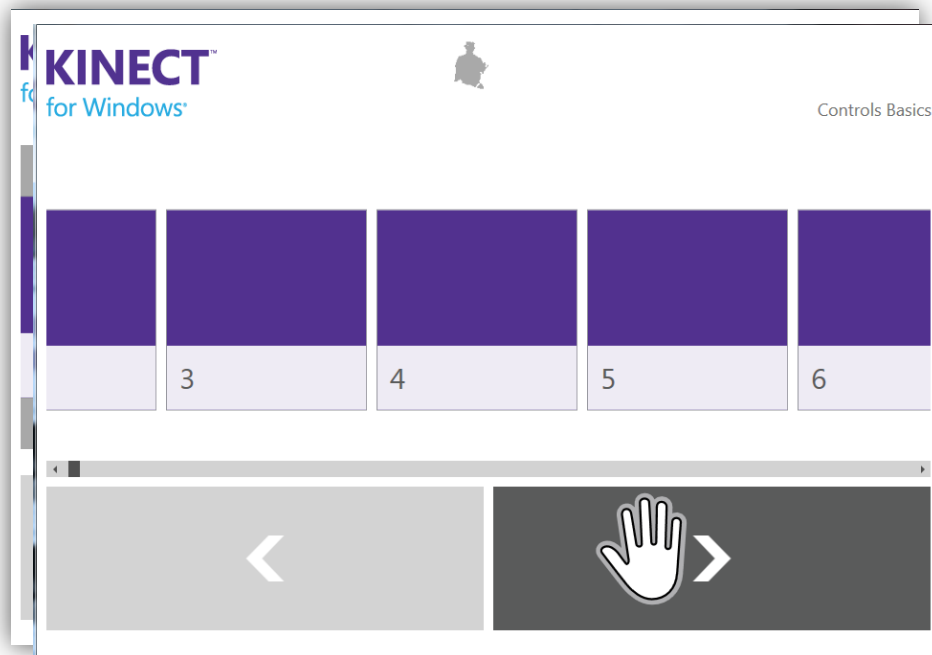


図14：ユーザービューウィンドウ

# Depth Basics

- センサーと対象の距離感をグレースケール化するプログラム。スクリーンショットも可能。
- 左下のチェックボックスで、深さの範囲を近い範囲に変えることが出来る。
- WPF,D2D,WPF-VB版有

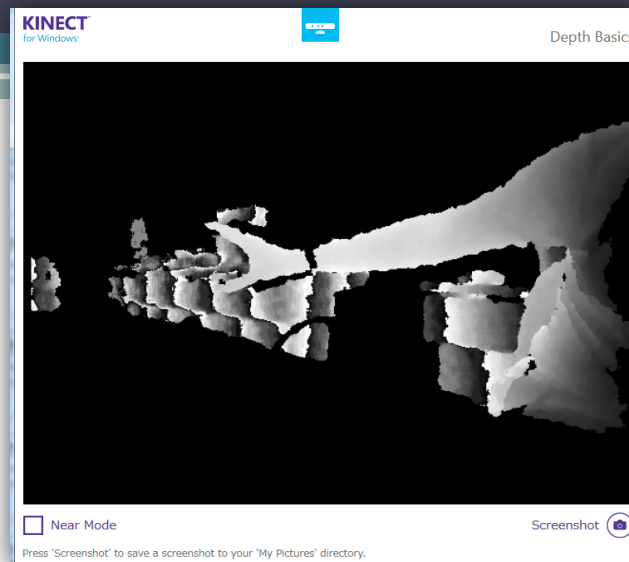


図15：グレースケール化1



図16：グレースケール化2

# Depth-D3D

- センサーと対象の距離感を  
グレースケール化するプログラム。  
対象を3D化して表現。  
スクリーンショットも可能。

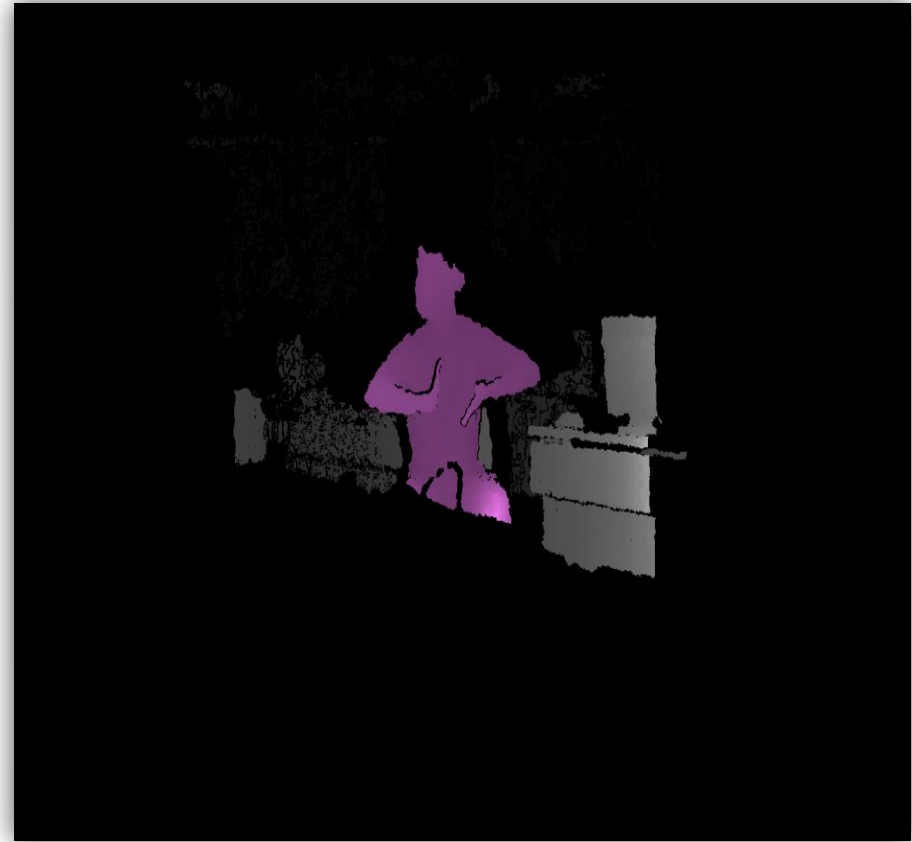


図17：グレースケール化(3D)



# Depth with Color-D3D

- センサーと対象の距離感を  
カラーで表現するプログラ  
ム。                      スクリーン  
ショットも可能。

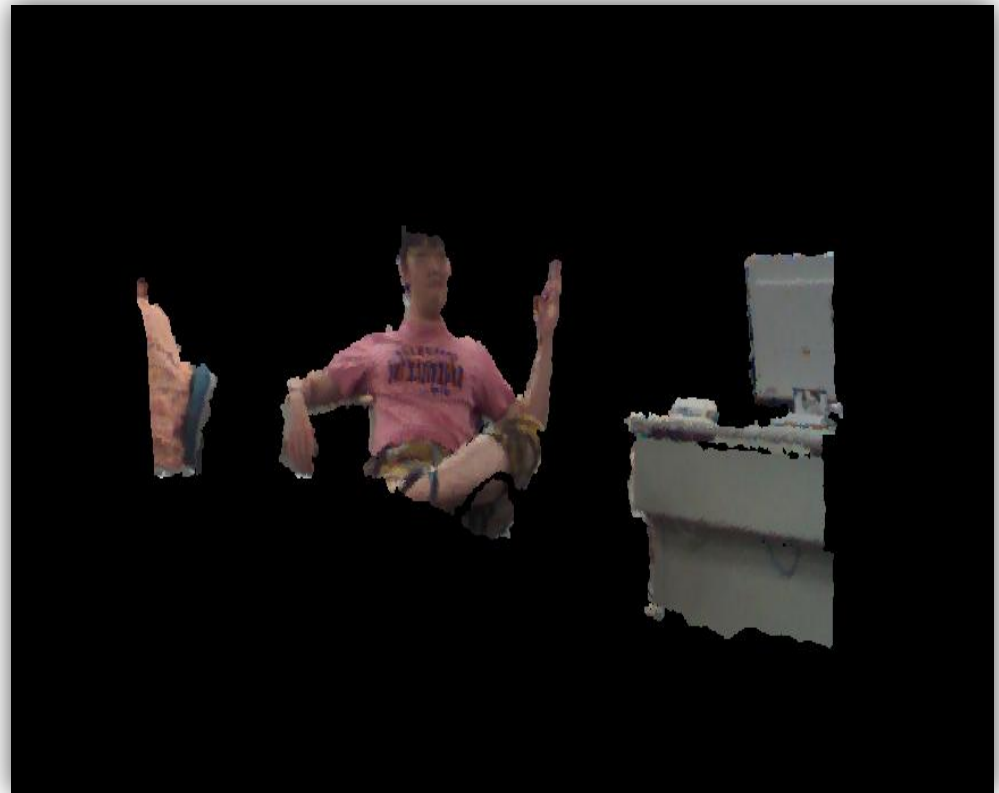


図18：カラー表現

# Face Tracking 3D-WPF

- 撮影対象の顔を認識し、マスクにしてレンダリング表示するプログラム。

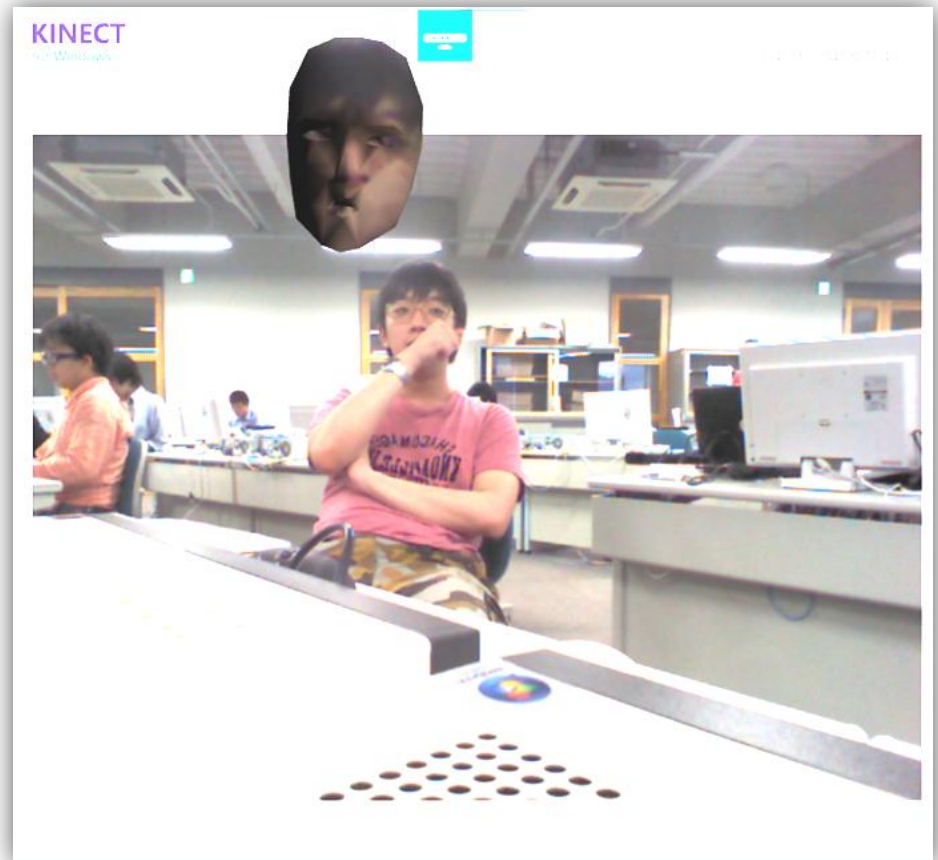


図19：撮影対象のマスクレンダリング

# Face Tracking Basics-WPF

- 顔を認識して、ポリゴンを生成し、顔の上に表示している。



図20：顔認識ポリゴン表示

# Face Tracking Visualization

- 訳: フェイストラッキングのSDKを使用し、ひとつの顔、または複数の顔をトラッキングし、視覚化するデモンストレーションです。
- ウィンドウを二分割し、右半分にはColor Image Streamで取得しているデータから、人の顔を認識し、ポリゴンデータを作成。
- 左半分は右側で作成された顔のポリゴンを元に簡易的な顔の描画を行っている。
- 顔の向きや口を開くなどの動作も左側の顔に反映される。

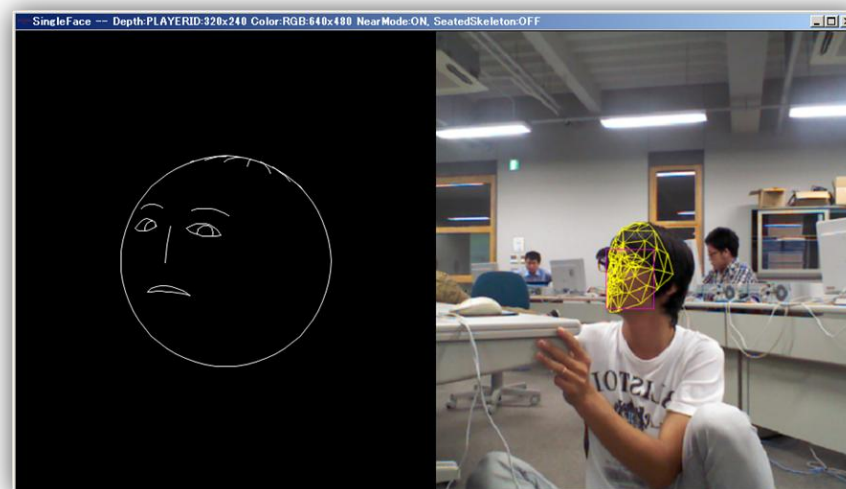


図21：顔認識と簡易描画

# Infrared Basics

- 訳:画面を毎秒30フレームで更新する基本的な赤外線カメラの機能を使ったデモンストレーションです。
- 赤外線カメラが取得したデータを30fpsで表示する。
- 実際に試すことはできなかったが、赤外線カメラを使っているので、暗闇でも同じように映るのではないかと予想される。
- D2D版とWPF版の二種類のサンプルがある。

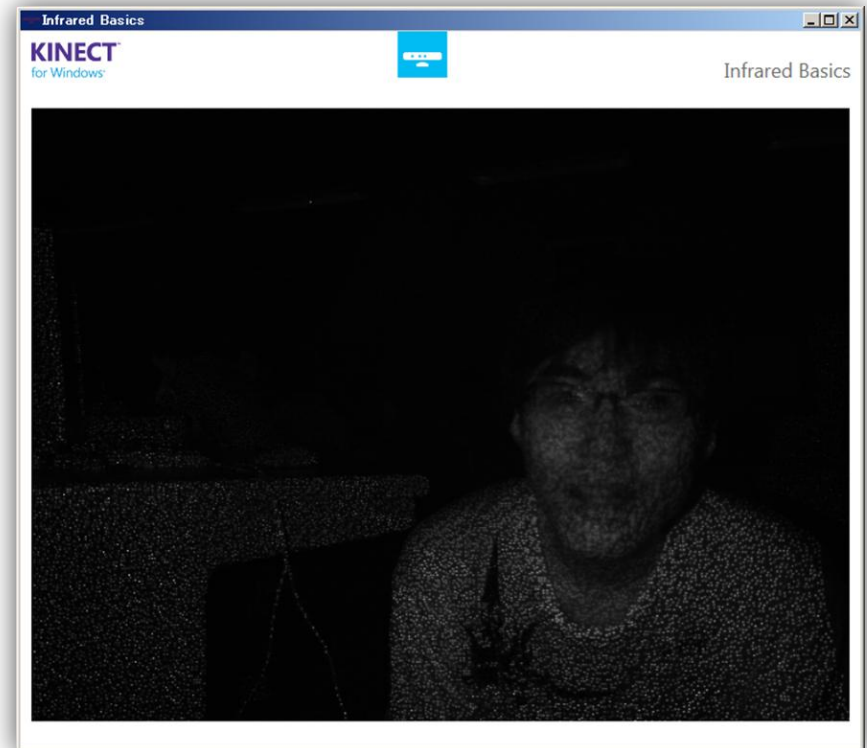


図22：赤外線カメラ

# Interaction Gallery-WPF

- 訳: ナビゲーションのための例で、係合、記事ビューアー、ビデオプレーヤー、HIG(Human Interface Guidelines)に記載された設計原理に基づいたpannablemapをKinectを用いてインタラクティブな総合的操作を体験できるデモンストレーション。
- Kinectが人の手を認識し、ポインタの様なものを表示。
- 手を突き出すことにより、画像や動画を開くなど、マウスの左クリックの様な挙動を得られた。

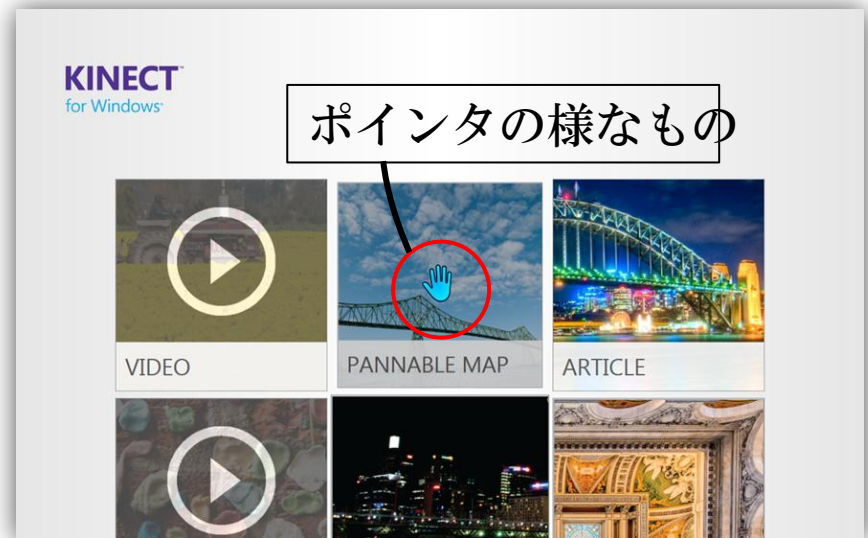


図23：ナビゲーションデモンストレーション



# Kinect Explorer

- 訳: KinectのColorImageStream、DepthImageStream、SkeletonStream、AudioSourceを使い方をデモンストレーションします。
- colorImageStream、DepthImageStream、SkeletonStream、AudioSourceの機能をGUI上で操作・切り替えが出来る。
- D2D版とWPF版の二種類のサンプルがある。



図24：操作・切り替え画面

# Shape Game

- スケルトントラッキングと音声認識を使用した簡単なゲームのデモンストレーションです。
- Kinectがトラッキングしたスケルトンをゲーム画面に表示し、画面上部から落ちてくる様々なオブジェクトをスケルトンで触れることでスコアを得ることができるゲーム。
- "Start", "Stop"など決められた単語を音声認識し、音声でもゲームを操作することができる。



図25：ゲーム画面



# Skeleton Basics-WPF

- Skeleton Streamの基本的な動作の確認
- 説明文から30fpsで更新されていることが分かる。
- Seated Modeにチェックを入れることで図2のように、肩から上だけが認識される。

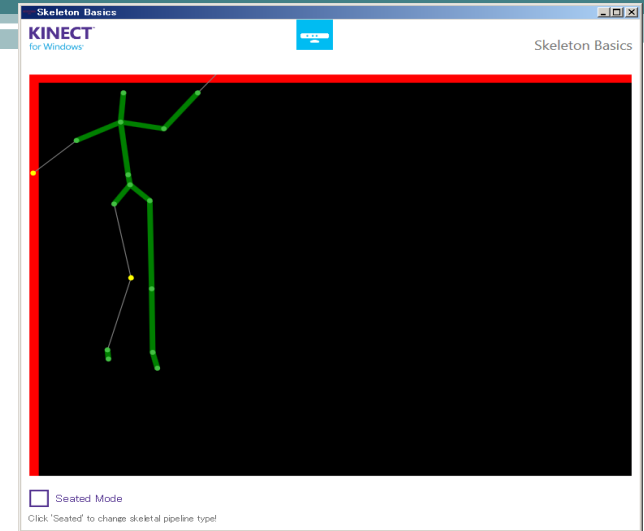


図26:Seated Mode off

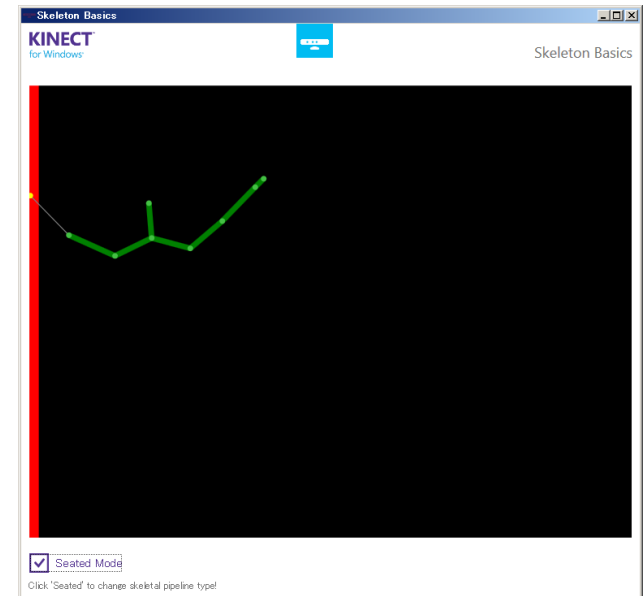


図27:Seated Mode on

# Speech Basics-WPF

- 音声認識でカメラが動く。
- 命令は全部で4種類ある。  
“Forward”, “Back”  
“Turn Left”, “Turn Right”

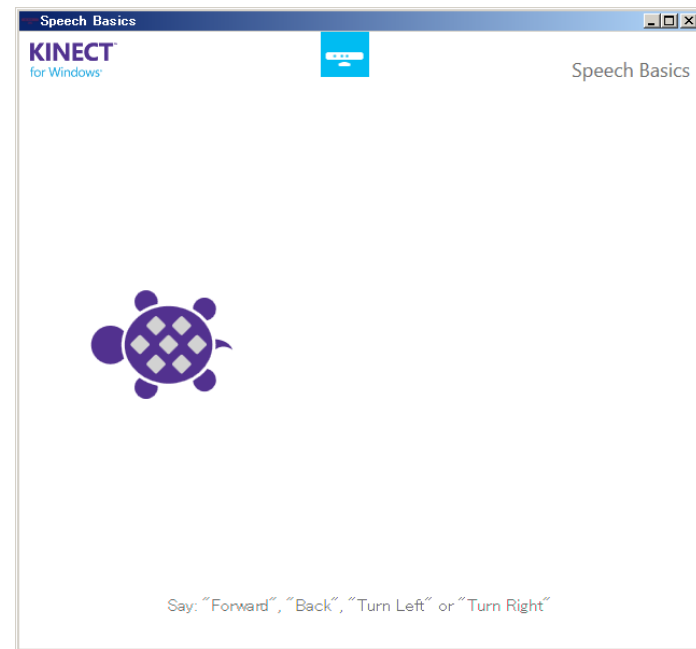


図28:真ん中から移動したカメラ

# Tic Tac Toe-WPF

- Kinectに入った人を2人まで認識して○×を割り当てる。
- 番号を英語で発音することでその時のターンのプレイヤーによって○×がその数字に入れられる。

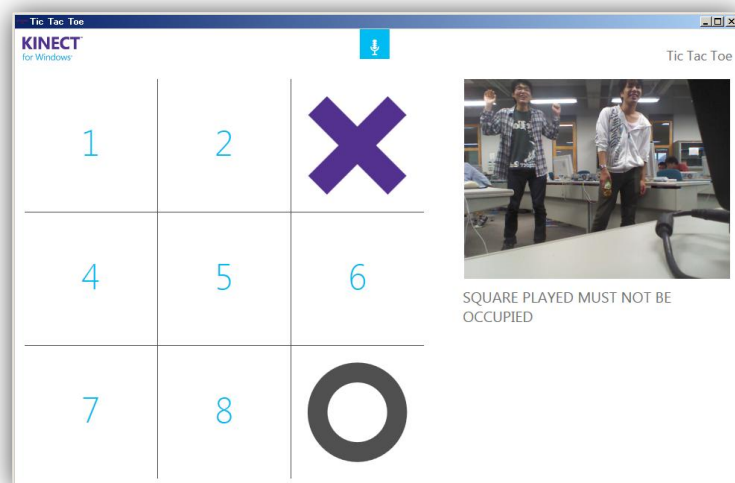


図29:○×ゲーム

# WPF D3D Interop

- Magnifierのオン/オフで虫眼鏡の表示/非表示ができる。
- Sizeで虫眼鏡の大きさ、Scaleで拡大の度合いを変更できる。
- Radiusをいじるとズームイン/アウトができる。
- Thetaは垂直軸回転、Phiは水平軸回転ができる。

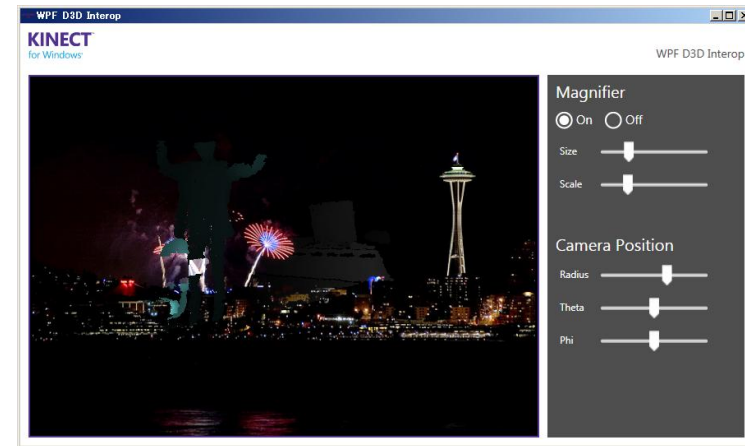


図30:Interopの実行結果

# XNA Basics

- 説明文からColor Image StreamとSkeleton StreamとDepth Image StreamのC#/XNAによるサンプルということが分かる。
- 図6はSkeleton Sampleで、図7はDepth Sampleである。

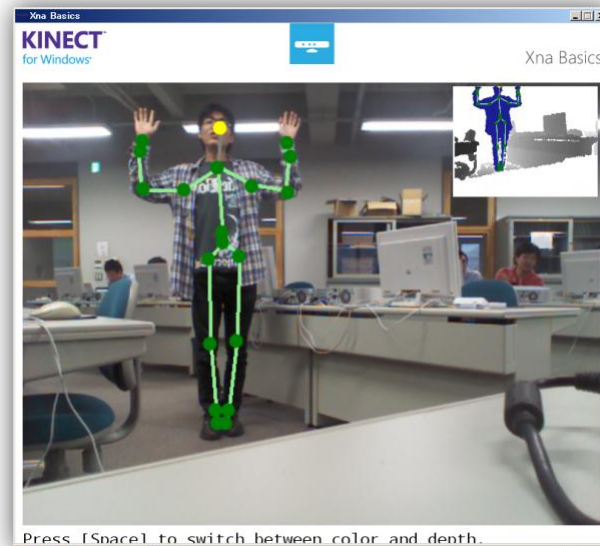


図31: Skeleton Sample

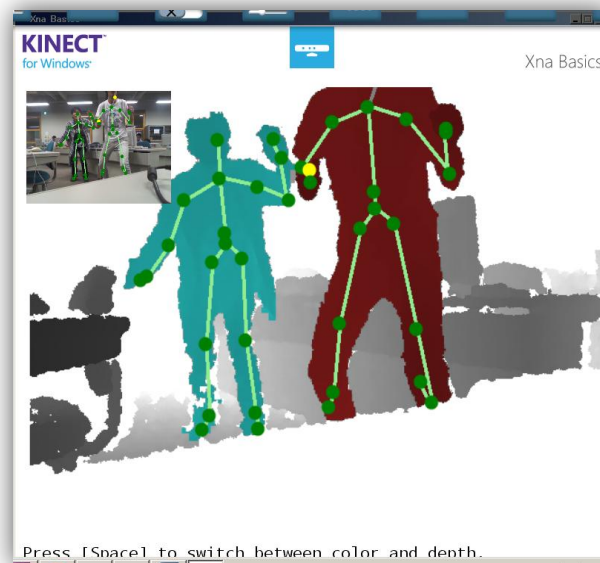


図32: Depth Sample

# 起動できなかったサンプル

- Kinect Bridge With Matlab Basics-D2D  
→Matlabを利用しているため
  - Kinect Bridge With OpenCV Basics-D2D  
→OpenCVを利用しているため
  - Kinect Fusion Basics-D2D(-WPF)
  - Kinect Fusion Color Basics-D2D(-WPF)
  - Kinect Fusion Explorer-D2D(-WPF)
  - Kinect Fusion Explorer Multi Static
  - Kinect Fusion Head Scanning-WPF
- 上記5つはDirectX11を利用しているため