Programming with the Kinect for Windows SDK
What we’ll cover

- Kinect Sensor
- Using Cameras
- Understanding Depth Data
- Skeletal Tracking
- Audio
RGB CAMERA

MULTI-ARRAY MIC

3D DEPTH SENSORS

MOTORIZED TILT
SDK Architecture

- **Video Components**
  - NU1 API
  - Video stream control
  - WinUSB camera stack

- **Audio Components**
  - Windows Core Audio and Speech APIs
  - DMO codec for mic array
  - Audio stream control
  - USBAudio audio stack

- **Kernel Mode**
  - Kernel-mode drivers for Kinect for Windows

- **User Mode**
  - User Mode

- **Hardware**
  - Motor
  - Cameras
  - Audio mic array

- **USB Hub**
  - Motor
  - Cameras
  - Audio mic array

- **Components**
  - Kinect for Windows SDK
  - Windows components
  - User-created components
SDK Architecture

Applications

Video Components
- NUI API

Audio Components
- Windows Core Audio and Speech APIs
- DMO codec for mic array

Kernel-mode drivers for Kinect for Windows

Device setup
- WinUSB device stack
- WinUSB camera stack

Device access
- USB Hub
- WinUSB audio stack
- USBAudio audio stack

Video stream control
- Video stream control

Audio stream control
- Audio stream control

Kernel Mode

User Mode

Hardware

Motor
- Camera setup
- Cameras
- Audio mic array

- Kinect sensor

- Kinect for Windows SDK
- Windows components
- User-created components
SDK Architecture

Video Components
- NUI API
- Video stream control
- WinUSB camera stack
- WinUSB device stack
- Kernel-mode drivers for Kinect for Windows

Audio Components
- Windows Core Audio and Speech APIs
- DMO codec for mic array
- Audio stream control
- USBAudio audio stack
- WinUSB camera stack

Hardware
- USB Hub
- Motor
- Cameras
- Audio mic array
- Kinect sensor

User Mode
- User Mode
- Application

Kernel Mode
- Kernel Mode
- System

User-created components
- Windows components
- User-created components

Windows components
SDK Architecture

Video Components
- NUI API

Audio Components
- Windows Core Audio and Speech APIs
- DMO codec for mic array

Hardware
- Kinect sensor
- Cameras
- Audio mic array
- Motor

Device setup
- Device access
- Video stream control
- Audio stream control

Kernel-mode drivers for Kinect for Windows

User Mode

Kernel Mode

User-mode created components
- USB Hub
- WinUSB device stack
- WinUSB camera stack
- USBAudio audio stack
Using Cameras

Demos
Understanding Depth Data

- ImageFrame.Image.Bits
- Array of bytes - `public byte[] Bits;`
- Array
  - Starts at top left of image
  - Moves left to right, then top to bottom
  - Represents distance for pixel in millimeters
Calculating Distance

- 2 bytes per pixel (16 bits)
- Depth – Distance per pixel
  - Bitshift **second byte by 8**
  - Distance \((0,0) = (\text{int})(\text{Bits}[0] \mid \text{Bits}[1] \ll 8);\)
  - VB \((\text{int})(\text{CInt}(\text{Bits}(0)) \text{ Or } \text{CInt}(\text{Bits}(1)) \ll 8);\)
- DepthAndPlayer Index – Includes Player index
  - Bitshift by **3 first byte** (player index), **5 second byte**
  - Distance \((0,0) = (\text{int})(\text{Bits}[0] \gg 3 \mid \text{Bits}[1] \ll 5);\)
  - VB:\((\text{int})(\text{CInt}(\text{Bits}(0)) \gg 3 \text{ Or } \text{CInt}(\text{Bits}(1)) \ll 5);\)
Depth Reference

- **Distance Range:** 850 mm to 4000 mm range
- **Depth value 0** means unknown
  - Shadows, low reflectivity, and high reflectivity among the few reasons
- **Player Index**
  - 0 – No player
  - 1 – Skeleton 0
  - 2 – Skeleton 1
  - ...
Demos
Skeleton Data
Joints

- Maximum two players tracked at once
  - Six player proposals
- Each player with set of \( <x, y, z> \) joints in meters
- Each joint has associated state
  - Tracked, Not tracked, or Inferred
- Inferred - Occluded, clipped, or low confidence joints
- Not Tracked - Rare, but your code must check for this state
Skeletal Tracking

SkeletonFrame
- Sealed Class
- Fields:
  - FloorClipPlane
  - FrameNumber
  - NormalToGravity
  - Quality
  - TimeStamp

SkeletonFrameReadyEventArgs
- Sealed Class
- Methods:
  - SkeletonFrameReadyEventArgs

SkeletonData
- Sealed Class
- Fields:
  - EnrollmentIndex
  - Position
  - Quality
  - TrackingID
  - TrackingState
  - UserID

Joint
- Struct
- Properties:
  - ID
  - Position
  - TrackingState
Demos
Audio Going Inside of Kinect

- Four microphone array with hardware-based audio processing
  - Multichannel echo cancellation (MEC)
  - Sound position tracking
  - Other digital signal processing (noise suppression and reduction)
Audio

- Kinect as a microphone
- Kinect for Speech Recognition
Multi-modal Feedback

- Engagement model
  - Mic indicator for speech-enabled menus
  - Keyword to engage

- Feedback and confirmation (both passive and active)
Speech Recognition

- Kinect Grammar available to download
- Grammar – What we are listening for
  - Code – GrammarBuilder, Choices
  - Speech Recognition Grammar Specification (SRGS)
    - C:\Program Files (x86)\Microsoft Speech Platform SDK\Samples\Sample Grammars\
Grammar

<!-- Confirmation_YesNo._value: string ["Yes", "No"] -->
<rule id="Confirmation_YesNo" scope="public">
  <example> yes </example>
  <example> no </example>
  <one-of>
    <item><ruleref uri="#Confirmation_Yes" /></item>
    <item><ruleref uri="#Confirmation_No" /></item>
  </one-of>
  <tag> out = rules.latest() </tag>
</rule>

<!-- Confirmation_Yes._value: string ["Yes"] -->
<rule id="Confirmation_Yes" scope="public">
  <example> yes </example>
  <example> yes please </example>
  <one-of>
    <item>yes</item>
    <item>yeah</item>
    <item>yep</item>
    <item>ok</item>
  </one-of>
  <item repeat="0-1"> please </item>
  <tag> out._value = "Yes";</tag>
</rule>
Demos
Resources

- Kinect Programming Walkthroughs
  - http://research.microsoft.com/kinectsdk/
- Coding4Fun Kinect Toolkit
  - http://c4fkinect.codeplex.com
- Kinect SDK Quickstarts
  - http://channel9.msdn.com/series/KinectSDKQuickstarts