

Study Kinect

1.0

Generated by Doxygen 1.12.0

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 AvatarController Class Reference	5
3.1.1 Detailed Description	6
3.1.2 Member Function Documentation	6
3.1.2.1 Kinect2AvatarPos()	6
3.1.2.2 Kinect2AvatarRot()	7
3.1.2.3 MapBones()	7
3.1.2.4 MoveAvatar()	7
3.1.2.5 SuccessfulCalibration()	7
3.1.2.6 TransformBone()	8
3.1.2.7 TransformSpecialBone()	8
3.1.2.8 UpdateAvatar()	8
3.1.3 Member Data Documentation	9
3.1.3.1 boneIndex2JointMap	9
3.1.3.2 boneIndex2MirrorJointMap	9
3.1.3.3 specIndex2JointMap	10
3.1.3.4 specIndex2MirrorJointMap	10
3.2 AvatarControllerClassic Class Reference	10
3.2.1 Detailed Description	12
3.2.2 Member Function Documentation	12
3.2.2.1 MapBones()	12
3.3 KinectWrapper.ColorBuffer Struct Reference	13
3.4 KinectWrapper.ColorCust Struct Reference	13
3.5 KinectWrapper.DepthBuffer Struct Reference	13
3.6 KinectGestures.GestureData Struct Reference	13
3.6.1 Detailed Description	14
3.7 KinectGestures.GestureListenerInterface Interface Reference	14
3.7.1 Member Function Documentation	14
3.7.1.1 GestureCancelled()	14
3.7.1.2 GestureCompleted()	15
3.7.1.3 GestureInProgress()	15
3.7.1.4 UserDetected()	16
3.7.1.5 UserLost()	16
3.8 KinectWrapper.INuiFrameTexture Interface Reference	16
3.9 KinectGestures Class Reference	17
3.9.1 Detailed Description	17
3.9.2 Member Function Documentation	17

3.9.2.1 CheckForGesture()	17
3.10 KinectManager Class Reference	18
3.10.1 Detailed Description	20
3.10.2 Member Function Documentation	20
3.10.2.1 DetectGesture()	20
3.10.2.2 GetDepthForPixel()	20
3.10.2.3 GetPlayer1ID()	21
3.10.2.4 GetPlayer2ID()	21
3.10.2.5 GetRawDepthMap()	21
3.10.2.6 GetUserPosition()	21
3.10.2.7 IsPlayerCalibrated()	21
3.10.2.8 IsUserDetected()	22
3.11 KinectWrapper Class Reference	22
3.11.1 Detailed Description	24
3.11.2 Member Function Documentation	24
3.11.2.1 GetNuiErrorString()	24
3.11.2.2 GetSkeletonJointOrientation()	25
3.11.2.3 GetSkeletonJointParent()	25
3.11.2.4 GetSkeletonMirroredJoint()	25
3.11.2.5 MapSkeletonPointToDepthPoint()	25
3.11.2.6 PollColor()	26
3.11.2.7 PollDepth()	26
3.11.2.8 PollSkeleton()	26
3.12 KinectWrapper.NuiImageBuffer Class Reference	27
3.13 KinectWrapper.NuiImageFrame Struct Reference	27
3.14 KinectWrapper.NuiImageViewArea Struct Reference	28
3.15 KinectWrapper.NuiLockedRect Struct Reference	28
3.16 KinectWrapper.NuiSkeletonBoneOrientation Struct Reference	28
3.17 KinectWrapper.NuiSkeletonBoneRotation Struct Reference	28
3.18 KinectWrapper.NuiSkeletonData Struct Reference	29
3.19 KinectWrapper.NuiSkeletonFrame Struct Reference	29
3.20 KinectWrapper.NuiSurfaceDesc Struct Reference	29
3.21 KinectWrapper.NuiTransformSmoothParameters Struct Reference	29

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

KinectWrapper.ColorBuffer	13
KinectWrapper.ColorCust	13
KinectWrapper.DepthBuffer	13
KinectGestures.GestureData	13
KinectGestures.GestureListenerInterface	14
KinectWrapper.INuiFrameTexture	16
KinectGestures	17
KinectWrapper	22
MonoBehaviour	
AvatarController	5
AvatarControllerClassic	10
KinectManager	18
KinectWrapper.NuiImageBuffer	27
KinectWrapper.NuiImageFrame	27
KinectWrapper.NuiImageViewArea	28
KinectWrapper.NuiLockedRect	28
KinectWrapper.NuiSkeletonBoneOrientation	28
KinectWrapper.NuiSkeletonBoneRotation	28
KinectWrapper.NuiSkeletonData	29
KinectWrapper.NuiSkeletonFrame	29
KinectWrapper.NuiSurfaceDesc	29
KinectWrapper.NuiTransformSmoothParameters	29

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AvatarController	
AvatarController 클래스는 Kinect를 사용하여 아바타의 움직임과 회전을 제어합니다. 사용자의 스켈레톤 데이터에 기반하여 아바타의 각 뼈를 업데이트합니다	5
AvatarControllerClassic	
AvatarControllerClassic 클래스는 Kinect 데이터를 사용하여 아바타의 뼈대를 매핑하는 기능을 제공합니다	10
KinectWrapper.ColorBuffer	13
KinectWrapper.ColorCust	13
KinectWrapper.DepthBuffer	13
KinectGestures.GestureData	
제스처 데이터 구조체로, 각 제스처의 상태를 저장합니다	13
KinectGestures.GestureListenerInterface	14
KinectWrapper.INuiFrameTexture	16
KinectGestures	
KinectGestures 클래스는 Kinect 센서를 사용하여 사용자 제스처를 감지하는 기능을 제공합니다	17
KinectManager	
KinectManager 클래스는 Kinect v1 센서를 이용하여 사용자 스켈레톤 데이터를 관리하고, 이를 기반으로 아바타의 움직임 및 제스처를 처리하는 기능을 제공합니다	18
KinectWrapper	
KinectWrapper 클래스는 Kinect 센서와 상호작용하기 위한 메소드와 구조체를 포함합니다	22
KinectWrapper.NuiImageBuffer	27
KinectWrapper.NuiImageFrame	27
KinectWrapper.NuiImageViewArea	28
KinectWrapper.NuiLockedRect	28
KinectWrapper.NuiSkeletonBoneOrientation	28
KinectWrapper.NuiSkeletonBoneRotation	28
KinectWrapper.NuiSkeletonData	29
KinectWrapper.NuiSkeletonFrame	29
KinectWrapper.NuiSurfaceDesc	29
KinectWrapper.NuiTransformSmoothParameters	29

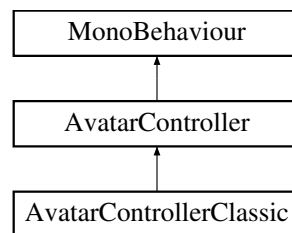
Chapter 3

Class Documentation

3.1 AvatarController Class Reference

[AvatarController](#) 클래스는 Kinect를 사용하여 아바타의 움직임과 회전을 제어합니다. 사용자의 스켈레톤 데이터에 기반하여 아바타의 각 뼈를 업데이트합니다.

Inheritance diagram for AvatarController:



Public Member Functions

- void **Awake** ()
Awake 메소드는 아바타의 초기 설정을 수행합니다. 뼈를 매핑하고 초기 회전을 가져옵니다.
- void **UpdateAvatar** (uint UserID)
UpdateAvatar 메소드는 매 프레임마다 아바타를 업데이트합니다. 사용자의 스켈레톤 데이터에 따라 아바타의 위치와 회전을 조정합니다.
- void **ResetToInitialPosition** ()
ResetToInitialPosition 메소드는 아바타의 뼈를 초기 위치와 회전으로 리셋합니다.
- void **SuccessfulCalibration** (uint userId)
SuccessfulCalibration 메소드는 사용자가 성공적으로 보정되었을 때 호출됩니다. 아바타의 위치를 리셋하고 오프셋을 재보정합니다.

Public Attributes

- bool **mirroredMovement** = false
- bool **verticalMovement** = false
- float **smoothFactor** = 5f
- bool **offsetRelativeToSensor** = false

Protected Member Functions

- void [TransformBone](#) (uint userId, KinectWrapper.NuiSkeletonPositionIndex joint, int boneIndex, bool flip)
TransformBone 메소드는 Kinect에서 추적된 관절 회전을 아바타의 뼈에 적용합니다.
- void [TransformSpecialBone](#) (uint userId, KinectWrapper.NuiSkeletonPositionIndex joint, KinectWrapper.NuiSkeletonPositionIndex jointParent, int boneIndex, Vector3 baseDir, bool flip)
TransformSpecialBone 메소드는 특수 관절에 대한 회전을 적용합니다.
- void [MoveAvatar](#) (uint UserID)
MoveAvatar 메소드는 아바타를 3D 공간에서 이동합니다. 사용자의 척추 위치를 기반으로 아바타의 루트를 이동합니다.
- virtual void [MapBones](#) ()
MapBones 메소드는 아바타의 뼈를 Kinect 관절에 매핑합니다.
- void [GetInitialRotations](#) ()
GetInitialRotations 메소드는 뼈의 초기 회전을 캡처합니다.
- Quaternion [Kinect2AvatarRot](#) (Quaternion jointRotation, int boneIndex)
Kinect2AvatarRot 메소드는 Kinect 관절 회전을 아바타 관절 회전으로 변환합니다.
- Vector3 [Kinect2AvatarPos](#) (Vector3 jointPosition, bool bMoveVertically)
Kinect2AvatarPos 메소드는 Kinect 위치를 아바타 스켈레톤 위치로 변환합니다.

Protected Attributes

- int **moveRate** = 1
- Transform **bodyRoot**
- GameObject **offsetNode**
- Transform[] **bones**
- Quaternion[] **initialRotations**
- Quaternion[] **initialLocalRotations**
- Vector3 **initialPosition**
- Quaternion **initialRotation**
- bool **offsetCalibrated** = false
- float **xOffset**
- float **yOffset**
- float **zOffset**
- [KinectManager](#) **kinectManager**
- readonly Dictionary< int, KinectWrapper.NuiSkeletonPositionIndex > [boneIndex2JointMap](#)
- readonly Dictionary< int, List< KinectWrapper.NuiSkeletonPositionIndex > > [specIndex2JointMap](#)
- readonly Dictionary< int, KinectWrapper.NuiSkeletonPositionIndex > [boneIndex2MirrorJointMap](#)
- readonly Dictionary< int, List< KinectWrapper.NuiSkeletonPositionIndex > > [specIndex2MirrorJointMap](#)

Properties

- new Transform **transform** [get]

3.1.1 Detailed Description

[AvatarController](#) 클래스는 Kinect를 사용하여 아바타의 움직임과 회전을 제어합니다. 사용자의 스켈레톤 데이터에 기반하여 아바타의 각 뼈를 업데이트합니다.

3.1.2 Member Function Documentation

3.1.2.1 Kinect2AvatarPos()

```
Vector3 AvatarController.Kinect2AvatarPos (
    Vector3 jointPosition,
    bool bMoveVertically) [protected]
```

Kinect2AvatarPos 메소드는 Kinect 위치를 아바타 스켈레톤 위치로 변환합니다.

Parameters

<i>jointPosition</i>	Kinect 관절 위치
<i>bMoveVertically</i>	수직 이동 여부

Returns

변환된 아바타 위치

3.1.2.2 Kinect2AvatarRot()

```
Quaternion AvatarController.Kinect2AvatarRot (
    Quaternion jointRotation,
    int boneIndex) [protected]
```

Kinect2AvatarRot 메소드는 Kinect 관절 회전을 아바타 관절 회전으로 변환합니다.

Parameters

<i>jointRotation</i>	Kinect 관절 회전
<i>boneIndex</i>	변환할 뼈 인덱스

Returns

변환된 아바타 관절 회전

3.1.2.3 MapBones()

```
virtual void AvatarController.MapBones () [protected], [virtual]
```

MapBones 메소드는 아바타의 뼈를 Kinect 관절에 매핑합니다.

Reimplemented in [AvatarControllerClassic](#).

3.1.2.4 MoveAvatar()

```
void AvatarController.MoveAvatar (
    uint UserID) [protected]
```

MoveAvatar 메소드는 아바타를 3D 공간에서 이동합니다. 사용자의 척추 위치를 기반으로 아바타의 루트를 이동합니다.

Parameters

<i>UserID</i>	이동할 사용자 ID
---------------	------------

3.1.2.5 SuccessfulCalibration()

```
void AvatarController.SuccessfulCalibration (
    uint userId)
```

SuccessfulCalibration 메소드는 사용자가 성공적으로 보정되었을 때 호출됩니다. 아바타의 위치를 리셋하고 오프셋을 재보정합니다.

Parameters

<i>userId</i>	성공적으로 보정된 사용자 ID
---------------	------------------

3.1.2.6 TransformBone()

```
void AvatarController.TransformBone (
    uint userId,
    KinectWrapper.NuiSkeletonPositionIndex joint,
    int boneIndex,
    bool flip) [protected]
```

TransformBone 메소드는 Kinect에서 추적된 관절 회전을 아바타의 뼈에 적용합니다.

Parameters

<i>userId</i>	사용자 ID
<i>joint</i>	적용할 Kinect 관절
<i>boneIndex</i>	적용할 뼈 인덱스
<i>flip</i>	반전 여부

3.1.2.7 TransformSpecialBone()

```
void AvatarController.TransformSpecialBone (
    uint userId,
    KinectWrapper.NuiSkeletonPositionIndex joint,
    KinectWrapper.NuiSkeletonPositionIndex jointParent,
    int boneIndex,
    Vector3 baseDir,
    bool flip) [protected]
```

TransformSpecialBone 메소드는 특수 관절에 대한 회전을 적용합니다.

Parameters

<i>userId</i>	사용자 ID
<i>joint</i>	적용할 Kinect 관절
<i>jointParent</i>	부모 관절
<i>boneIndex</i>	적용할 뼈 인덱스
<i>baseDir</i>	기본 방향
<i>flip</i>	반전 여부

3.1.2.8 UpdateAvatar()

```
void AvatarController.UpdateAvatar (
    uint UserID)
```

UpdateAvatar 메소드는 매 프레임마다 아바타를 업데이트합니다. 사용자의 스켈레톤 데이터에 따라 아바타의 위치와 회전을 조정합니다.

Parameters

<i>UserID</i>	업데이트할 사용자 ID
---------------	--------------

3.1.3 Member Data Documentation

3.1.3.1 boneIndex2JointMap

readonly Dictionary<int, KinectWrapper.NuiSkeletonPositionIndex> AvatarController.boneIndex2↔
JointMap [protected]

Initial value:

```
= new Dictionary<int, KinectWrapper.NuiSkeletonPositionIndex>
{
    {0, KinectWrapper.NuiSkeletonPositionIndex.HipCenter},
    {1, KinectWrapper.NuiSkeletonPositionIndex.Spine},
    {2, KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter},
    {3, KinectWrapper.NuiSkeletonPositionIndex.Head},

    {5, KinectWrapper.NuiSkeletonPositionIndex.ShoulderLeft},
    {6, KinectWrapper.NuiSkeletonPositionIndex.ElbowLeft},
    {7, KinectWrapper.NuiSkeletonPositionIndex.WristLeft},
    {8, KinectWrapper.NuiSkeletonPositionIndex.HandLeft},

    {10, KinectWrapper.NuiSkeletonPositionIndex.ShoulderRight},
    {11, KinectWrapper.NuiSkeletonPositionIndex.ElbowRight},
    {12, KinectWrapper.NuiSkeletonPositionIndex.WristRight},
    {13, KinectWrapper.NuiSkeletonPositionIndex.HandRight},

    {14, KinectWrapper.NuiSkeletonPositionIndex.HipLeft},
    {15, KinectWrapper.NuiSkeletonPositionIndex.KneeLeft},
    {16, KinectWrapper.NuiSkeletonPositionIndex.AnkleLeft},
    {17, KinectWrapper.NuiSkeletonPositionIndex.FootLeft},

    {18, KinectWrapper.NuiSkeletonPositionIndex.HipRight},
    {19, KinectWrapper.NuiSkeletonPositionIndex.KneeRight},
    {20, KinectWrapper.NuiSkeletonPositionIndex.AnkleRight},
    {21, KinectWrapper.NuiSkeletonPositionIndex.FootRight},
}
```

3.1.3.2 boneIndex2MirrorJointMap

readonly Dictionary<int, KinectWrapper.NuiSkeletonPositionIndex> AvatarController.boneIndex2↔
MirrorJointMap [protected]

Initial value:

```
= new Dictionary<int, KinectWrapper.NuiSkeletonPositionIndex>
{
    {0, KinectWrapper.NuiSkeletonPositionIndex.HipCenter},
    {1, KinectWrapper.NuiSkeletonPositionIndex.Spine},
    {2, KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter},
    {3, KinectWrapper.NuiSkeletonPositionIndex.Head},

    {5, KinectWrapper.NuiSkeletonPositionIndex.ShoulderRight},
    {6, KinectWrapper.NuiSkeletonPositionIndex.ElbowRight},
    {7, KinectWrapper.NuiSkeletonPositionIndex.WristRight},
    {8, KinectWrapper.NuiSkeletonPositionIndex.HandRight},

    {10, KinectWrapper.NuiSkeletonPositionIndex.ShoulderLeft},
    {11, KinectWrapper.NuiSkeletonPositionIndex.ElbowLeft},
    {12, KinectWrapper.NuiSkeletonPositionIndex.WristLeft},
    {13, KinectWrapper.NuiSkeletonPositionIndex.HandLeft},

    {14, KinectWrapper.NuiSkeletonPositionIndex.HipRight},
    {15, KinectWrapper.NuiSkeletonPositionIndex.KneeRight},
    {16, KinectWrapper.NuiSkeletonPositionIndex.AnkleRight},
    {17, KinectWrapper.NuiSkeletonPositionIndex.FootRight},

    {18, KinectWrapper.NuiSkeletonPositionIndex.HipLeft},
    {19, KinectWrapper.NuiSkeletonPositionIndex.KneeLeft},
    {20, KinectWrapper.NuiSkeletonPositionIndex.AnkleLeft},
    {21, KinectWrapper.NuiSkeletonPositionIndex.FootLeft},
}
```

3.1.3.3 specIndex2JointMap

readonly Dictionary<int, List<KinectWrapper.NuiSkeletonPositionIndex> > AvatarController.↵
specIndex2JointMap [protected]

Initial value:

```
= new Dictionary<int, List<KinectWrapper.NuiSkeletonPositionIndex>
{
    {4, new List<KinectWrapper.NuiSkeletonPositionIndex>
    {KinectWrapper.NuiSkeletonPositionIndex.ShoulderLeft,
    KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter} },
    {9, new List<KinectWrapper.NuiSkeletonPositionIndex>
    {KinectWrapper.NuiSkeletonPositionIndex.ShoulderRight,
    KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter} },
}
```

3.1.3.4 specIndex2MirrorJointMap

readonly Dictionary<int, List<KinectWrapper.NuiSkeletonPositionIndex> > AvatarController.↵
specIndex2MirrorJointMap [protected]

Initial value:

```
= new Dictionary<int, List<KinectWrapper.NuiSkeletonPositionIndex>
{
    {4, new List<KinectWrapper.NuiSkeletonPositionIndex>
    {KinectWrapper.NuiSkeletonPositionIndex.ShoulderRight,
    KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter} },
    {9, new List<KinectWrapper.NuiSkeletonPositionIndex>
    {KinectWrapper.NuiSkeletonPositionIndex.ShoulderLeft,
    KinectWrapper.NuiSkeletonPositionIndex.ShoulderCenter} },
}
```

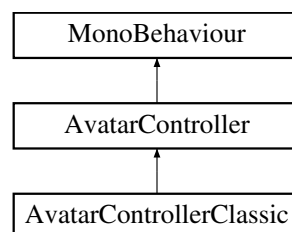
The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/AvatarController.cs

3.2 AvatarControllerClassic Class Reference

[AvatarControllerClassic](#) 클래스는 Kinect 데이터를 사용하여 아바타의 뼈대를 매핑하는 기능을 제공합니다.

Inheritance diagram for AvatarControllerClassic:



Public Attributes

- Transform **HipCenter**
- Transform **Spine**
- Transform **Neck**
- Transform **Head**
- Transform **LeftClavicle**
- Transform **LeftUpperArm**
- Transform **LeftElbow**
- Transform **LeftHand**
- Transform **RightClavicle**
- Transform **RightUpperArm**
- Transform **RightElbow**
- Transform **RightHand**
- Transform **LeftThigh**
- Transform **LeftKnee**
- Transform **LeftFoot**
- Transform **RightThigh**
- Transform **RightKnee**
- Transform **RightFoot**
- Transform **BodyRoot**
- GameObject **OffsetNode**

Public Attributes inherited from AvatarController

- bool **mirroredMovement** = false
- bool **verticalMovement** = false
- float **smoothFactor** = 5f
- bool **offsetRelativeToSensor** = false

Protected Member Functions

- override void **MapBones** ()
뼈대를 아바타 모델에 매핑합니다.

Protected Member Functions inherited from AvatarController

- void **TransformBone** (uint userId, KinectWrapper.NuiSkeletonPositionIndex joint, int boneIndex, bool flip)
TransformBone 메소드는 Kinect에서 추적된 관절 회전을 아바타의 뼈에 적용합니다.
- void **TransformSpecialBone** (uint userId, KinectWrapper.NuiSkeletonPositionIndex joint, KinectWrapper.NuiSkeletonPositionIndex jointParent, int boneIndex, Vector3 baseDir, bool flip)
TransformSpecialBone 메소드는 특수 관절에 대한 회전을 적용합니다.
- void **MoveAvatar** (uint UserID)
MoveAvatar 메소드는 아바타를 3D 공간에서 이동합니다. 사용자의 척추 위치를 기반으로 아바타의 루트를 이동합니다.
- void **GetInitialRotations** ()
GetInitialRotations 메소드는 뼈의 초기 회전을 캡처합니다.
- Quaternion **Kinect2AvatarRot** (Quaternion jointRotation, int boneIndex)
Kinect2AvatarRot 메소드는 Kinect 관절 회전을 아바타 관절 회전으로 변환합니다.
- Vector3 **Kinect2AvatarPos** (Vector3 jointPosition, bool bMoveVertically)
Kinect2AvatarPos 메소드는 Kinect 위치를 아바타 스켈레톤 위치로 변환합니다.

Additional Inherited Members

Public Member Functions inherited from [AvatarController](#)

- void **Awake** ()
Awake 메소드는 아바타의 초기 설정을 수행합니다. 뼈를 매핑하고 초기 회전을 가져옵니다.
- void **UpdateAvatar** (uint UserID)
UpdateAvatar 메소드는 매 프레임마다 아바타를 업데이트합니다. 사용자의 스켈레톤 데이터에 따라 아바타의 위치와 회전을 조정합니다.
- void **ResetToInitialPosition** ()
ResetToInitialPosition 메소드는 아바타의 뼈를 초기 위치와 회전으로 리셋합니다.
- void **SuccessfulCalibration** (uint userId)
SuccessfulCalibration 메소드는 사용자가 성공적으로 보정되었을 때 호출됩니다. 아바타의 위치를 리셋하고 오프셋을 재보정합니다.

Protected Attributes inherited from [AvatarController](#)

- int **moveRate** = 1
- Transform **bodyRoot**
- GameObject **offsetNode**
- Transform[] **bones**
- Quaternion[] **initialRotations**
- Quaternion[] **initialLocalRotations**
- Vector3 **initialPosition**
- Quaternion **initialRotation**
- bool **offsetCalibrated** = false
- float **xOffset**
- float **yOffset**
- float **zOffset**
- [KinectManager](#) **kinectManager**
- readonly Dictionary< int, KinectWrapper.NuiSkeletonPositionIndex > **boneIndex2JointMap**
- readonly Dictionary< int, List< KinectWrapper.NuiSkeletonPositionIndex > > **specIndex2JointMap**
- readonly Dictionary< int, KinectWrapper.NuiSkeletonPositionIndex > **boneIndex2MirrorJointMap**
- readonly Dictionary< int, List< KinectWrapper.NuiSkeletonPositionIndex > > **specIndex2MirrorJointMap**

Properties inherited from [AvatarController](#)

- new Transform **transform** [get]

3.2.1 Detailed Description

[AvatarControllerClassic](#) 클래스는 Kinect 데이터를 사용하여 아바타의 뼈대를 매핑하는 기능을 제공합니다.

3.2.2 Member Function Documentation

3.2.2.1 MapBones()

```
override void AvatarControllerClassic.MapBones () [protected], [virtual]
```

뼈대를 아바타 모델에 매핑합니다.

Reimplemented from [AvatarController](#).

The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/AvatarControllerClassic.cs

3.3 KinectWrapper.ColorBuffer Struct Reference

Public Attributes

- [ColorCust\[\]](#) pixels

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.4 KinectWrapper.ColorCust Struct Reference

Public Attributes

- byte **b**
- byte **g**
- byte **r**
- byte **a**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.5 KinectWrapper.DepthBuffer Struct Reference

Public Attributes

- [ushort\[\]](#) pixels

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.6 KinectGestures.GestureData Struct Reference

제스처 데이터 구조체로, 각 제스처의 상태를 저장합니다.

Public Attributes

- uint **userId**
- [Gestures](#) **gesture**
- int **state**
- float **timestamp**
- int **joint**
- Vector3 **jointPos**
- Vector3 **screenPos**
- float **tagFloat**
- Vector3 **tagVector**
- Vector3 **tagVector2**
- float **progress**
- bool **complete**
- bool **cancelled**
- List< [Gestures](#) > **checkForGestures**
- float **startTrackingAtTime**

3.6.1 Detailed Description

제스처 데이터 구조체로, 각 제스처의 상태를 저장합니다.

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectGestures.cs

3.7 KinectGestures.GestureListenerInterface Interface Reference

Public Member Functions

- void [UserDetected](#) (uint userId, int userIndex)
새로운 사용자가 감지되고 추적이 시작될 때 호출됩니다.
- void [UserLost](#) (uint userId, int userIndex)
사용자가 감지되지 않을 때 호출됩니다.
- void [GestureInProgress](#) (uint userId, int userIndex, [Gestures](#) gesture, float progress, KinectWrapper.NuiSkeletonPositionIndex joint, Vector3 screenPos)
제스처가 진행 중일 때 호출됩니다.
- bool [GestureCompleted](#) (uint userId, int userIndex, [Gestures](#) gesture, KinectWrapper.NuiSkeletonPositionIndex joint, Vector3 screenPos)
제스처가 완료되었을 때 호출됩니다.
- bool [GestureCancelled](#) (uint userId, int userIndex, [Gestures](#) gesture, KinectWrapper.NuiSkeletonPositionIndex joint)
제스처가 취소되었을 때 호출됩니다.

3.7.1 Member Function Documentation

3.7.1.1 GestureCancelled()

```
bool KinectGestures.GestureListenerInterface.GestureCancelled (
    uint userId,
    int userIndex,
    Gestures gesture,
    KinectWrapper.NuiSkeletonPositionIndex joint)
```

제스처가 취소되었을 때 호출됩니다.

Parameters

<i>userId</i>	사용자 ID
<i>userIndex</i>	사용자 인덱스
<i>gesture</i>	제스처 종류
<i>joint</i>	관절 위치

Returns

제스처 감지를 재시작해야 할지 여부

3.7.1.2 GestureCompleted()

```
bool KinectGestures.GestureListenerInterface.GestureCompleted (
    uint userId,
    int userIndex,
    Gestures gesture,
    KinectWrapper.NuiSkeletonPositionIndex joint,
    Vector3 screenPos)
```

제스처가 완료되었을 때 호출됩니다.

Parameters

<i>userId</i>	사용자 ID
<i>userIndex</i>	사용자 인덱스
<i>gesture</i>	제스처 종류
<i>joint</i>	관절 위치
<i>screenPos</i>	화면 위치

Returns

제스처 감지를 재시작해야 할지 여부

3.7.1.3 GestureInProgress()

```
void KinectGestures.GestureListenerInterface.GestureInProgress (
    uint userId,
    int userIndex,
    Gestures gesture,
    float progress,
    KinectWrapper.NuiSkeletonPositionIndex joint,
    Vector3 screenPos)
```

제스처가 진행 중일 때 호출됩니다.

Parameters

<i>userId</i>	사용자 ID
<i>userIndex</i>	사용자 인덱스
<i>gesture</i>	제스처 종류
<i>progress</i>	제스처 진행 상태
<i>joint</i>	관절 위치
<i>screenPos</i>	화면 위치

3.7.1.4 UserDetected()

```
void KinectGestures.GestureListenerInterface.UserDetected (
    uint userId,
    int userIndex)
```

새로운 사용자가 감지되고 추적이 시작될 때 호출됩니다.

Parameters

<i>userId</i>	사용자 ID
<i>userIndex</i>	사용자 인덱스

3.7.1.5 UserLost()

```
void KinectGestures.GestureListenerInterface.UserLost (
    uint userId,
    int userIndex)
```

사용자가 감지되지 않을 때 호출됩니다.

Parameters

<i>userId</i>	사용자 ID
<i>userIndex</i>	사용자 인덱스

The documentation for this interface was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectGestures.cs

3.8 KinectWrapper.INuiFrameTexture Interface Reference**Public Member Functions**

- int **BufferLen** ()
- int **Pitch** ()
- int **LockRect** (uint Level, ref [NuiLockedRect](#) pLockedRect, IntPtr pRect, uint Flags)
- int **GetLevelDesc** (uint Level, ref [NuiSurfaceDesc](#) pDesc)
- int **UnlockRect** (uint Level)

The documentation for this interface was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.9 KinectGestures Class Reference

[KinectGestures](#) 클래스는 Kinect 센서를 사용하여 사용자 제스처를 감지하는 기능을 제공합니다.

Classes

- struct [GestureData](#)
제스처 데이터 구조체로, 각 제스처의 상태를 저장합니다.
- interface [GestureListenerInterface](#)

Public Types

- enum [Gestures](#) {
None = 0, RaiseRightHand, RaiseLeftHand, Psi,
Tpose, Stop, Wave, Click,
SwipeLeft, SwipeRight, SwipeUp, SwipeDown,
RightHandCursor, LeftHandCursor, ZoomOut, ZoomIn,
Wheel, Jump, Squat, Push,
Pull }
감지할 제스처 종류를 정의합니다.

Static Public Member Functions

- static void [CheckForGesture](#) (uint userId, ref [GestureData](#) gestureData, float timestamp, ref Vector3[] jointsPos, ref bool[] jointsTracked)
제스처 상태를 확인합니다.

3.9.1 Detailed Description

[KinectGestures](#) 클래스는 Kinect 센서를 사용하여 사용자 제스처를 감지하는 기능을 제공합니다.

3.9.2 Member Function Documentation

3.9.2.1 CheckForGesture()

```
static void KinectGestures.CheckForGesture (
    uint userId,
    ref GestureData gestureData,
    float timestamp,
    ref Vector3[] jointsPos,
    ref bool[] jointsTracked) [static]
```

제스처 상태를 확인합니다.

Parameters

<i>userId</i>	사용자 ID
<i>gestureData</i>	제스처 데이터
<i>timestamp</i>	현재 타임스탬프
<i>jointsPos</i>	관절 위치 배열
<i>jointsTracked</i>	관절 추적 상태 배열

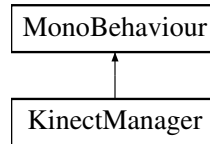
The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectGestures.cs

3.10 KinectManager Class Reference

KinectManager 클래스는 Kinect v1 센서를 이용하여 사용자 스켈레톤 데이터를 관리하고, 이를 기반으로 아바타의 움직임 및 제스처를 처리하는 기능을 제공합니다.

Inheritance diagram for KinectManager:



Public Types

- enum **Smoothing** : int { **None** , **Default** , **Medium** , **Aggressive** }

Public Member Functions

- bool **IsInitialized** ()
- ushort[] **GetRawDepthMap** ()
GetRawDepthMap 메소드는 원시 깊이/사용자 데이터를 반환합니다. *ComputeUserMap*이 *true*일 때만 유효합니다.
- ushort **GetDepthForPixel** (int x, int y)
GetDepthForPixel 메소드는 특정 픽셀에 대한 깊이 데이터를 반환합니다. *ComputeUserMap*이 *true*일 때만 유효합니다.
- Vector2 **GetDepthMapPosForJointPos** (Vector3 posJoint)
- Vector2 **GetColorMapPosForDepthPos** (Vector2 posDepth)
- Texture2D **GetUsersLblTex** ()
- Texture2D **GetUsersClrTex** ()
- bool **IsUserDetected** ()
IsUserDetected 메소드는 최소한 하나의 사용자가 감지되었는지 확인합니다.
- uint **GetPlayer1ID** ()
GetPlayer1ID 메소드는 *Player 1*의 *UserID*를 반환합니다.
- uint **GetPlayer2ID** ()
GetPlayer2ID 메소드는 *Player 2*의 *UserID*를 반환합니다.
- int **GetPlayer1Index** ()
- int **GetPlayer2Index** ()
- bool **IsPlayerCalibrated** (uint UserId)
IsPlayerCalibrated 메소드는 주어진 사용자가 보정되었는지 확인합니다.
- Vector3 **GetRawSkeletonJointPos** (uint UserId, int joint)
- Vector3 **GetUserPosition** (uint UserId)
GetUserPosition 메소드는 주어진 사용자의 위치를 반환합니다.
- Quaternion **GetUserOrientation** (uint UserId, bool flip)
- bool **IsJointTracked** (uint UserId, int joint)
- Vector3 **GetJointPosition** (uint UserId, int joint)
- Vector3 **GetJointLocalPosition** (uint UserId, int joint)
- Quaternion **GetJointOrientation** (uint UserId, int joint, bool flip)
- Quaternion **GetJointLocalOrientation** (uint UserId, int joint, bool flip)
- Vector3 **GetDirectionBetweenJoints** (uint UserId, int baseJoint, int nextJoint, bool flipX, bool flipZ)
- void **DetectGesture** (uint UserId, **KinectGestures.Gestures** gesture)

DetectGesture 메소드는 특정 제스처를 감지하고 이를 기반으로 다양한 동작을 실행합니다. 사용자의 제스처 진행률을 추적하고, 완료된 제스처에 대한 처리를 수행합니다.

- bool **ResetGesture** (uint UserId, [KinectGestures.Gestures](#) gesture)
- void **ResetPlayerGestures** (uint UserId)
- bool **DeleteGesture** (uint UserId, [KinectGestures.Gestures](#) gesture)
- void **ClearGestures** (uint UserId)
- int **GetGesturesCount** (uint UserId)
- List< [KinectGestures.Gestures](#) > **GetGesturesList** (uint UserId)
- bool **IsGestureDetected** (uint UserId, [KinectGestures.Gestures](#) gesture)
- bool **IsGestureComplete** (uint UserId, [KinectGestures.Gestures](#) gesture, bool bResetOnComplete)
- bool **IsGestureCancelled** (uint UserId, [KinectGestures.Gestures](#) gesture)
- float **GetGestureProgress** (uint UserId, [KinectGestures.Gestures](#) gesture)
- Vector3 **GetGestureScreenPos** (uint UserId, [KinectGestures.Gestures](#) gesture)
- void **ResetGestureListeners** ()
- void **ResetAvatarControllers** ()
- void **ClearKinectUsers** ()
- void **ResetFilters** ()

Static Public Member Functions

- static bool **IsKinectInitialized** ()
- static bool **IsCalibrationNeeded** ()

Public Attributes

- bool **TwoUsers** = false
- bool **ComputeUserMap** = false
- bool **ComputeColorMap** = false
- bool **DisplayUserMap** = false
- bool **DisplayColorMap** = false
- bool **DisplaySkeletonLines** = false
- float **DisplayMapsWidthPercent** = 20f
- float **SensorHeight** = 1.0f
- int **SensorAngle** = 0
- float **MinUserDistance** = 1.0f
- float **MaxUserDistance** = 0f
- bool **DetectClosestUser** = true
- bool **IgnoreInferredJoints** = true
- Smoothing **smoothing** = Smoothing.Default
- bool **UseBoneOrientationsFilter** = false
- bool **UseClippedLegsFilter** = false
- bool **UseBoneOrientationsConstraint** = true
- bool **UseSelfIntersectionConstraint** = false
- List< GameObject > **Player1Avatars**
- List< GameObject > **Player2Avatars**
- [KinectGestures.Gestures](#) **Player1CalibrationPose**
- [KinectGestures.Gestures](#) **Player2CalibrationPose**
- List< [KinectGestures.Gestures](#) > **Player1Gestures**
- List< [KinectGestures.Gestures](#) > **Player2Gestures**
- float **MinTimeBetweenGestures** = 0.7f
- List< MonoBehaviour > **GestureListeners**
- GUIText **CalibrationText**
- GameObject **HandCursor1**
- GameObject **HandCursor2**
- bool **ControlMouseCursor** = false
- GUIText **GesturesDebugText**
- List< [KinectGestures.GestureListenerInterface](#) > **gestureListeners**

Properties

- static [KinectManager Instance](#) [get]

3.10.1 Detailed Description

[KinectManager](#) 클래스는 Kinect v1 센서를 이용하여 사용자 스켈레톤 데이터를 관리하고, 이를 기반으로 아바타의 움직임 및 제스처를 처리하는 기능을 제공합니다.

3.10.2 Member Function Documentation

3.10.2.1 DetectGesture()

```
void KinectManager.DetectGesture (
    uint UserId,
    KinectGestures.Gestures gesture)
```

DetectGesture 메소드는 특정 제스처를 감지하고 이를 기반으로 다양한 동작을 실행합니다. 사용자의 제스처 진행률을 추적하고, 완료된 제스처에 대한 처리를 수행합니다.

Parameters

<i>UserId</i>	제스처를 감지할 사용자 ID
<i>gesture</i>	감지할 제스처 종류

3.10.2.2 GetDepthForPixel()

```
ushort KinectManager.GetDepthForPixel (
    int x,
    int y)
```

GetDepthForPixel 메소드는 특정 픽셀에 대한 깊이 데이터를 반환합니다. ComputeUserMap이 true일 때만 유효합니다.

Parameters

<i>x</i>	픽셀의 x좌표
<i>y</i>	픽셀의 y좌표

Returns

주어진 픽셀의 깊이 값

3.10.2.3 GetPlayer1ID()

```
uint KinectManager.GetPlayer1ID ()
```

GetPlayer1ID 메소드는 Player 1의 UserID를 반환합니다.

Returns

Player 1의 UserID 또는 0 (감지되지 않으면)

3.10.2.4 GetPlayer2ID()

```
uint KinectManager.GetPlayer2ID ()
```

GetPlayer2ID 메소드는 Player 2의 UserID를 반환합니다.

Returns

Player 2의 UserID 또는 0 (감지되지 않으면)

3.10.2.5 GetRawDepthMap()

```
ushort[] KinectManager.GetRawDepthMap ()
```

GetRawDepthMap 메소드는 원시 깊이/사용자 데이터를 반환합니다. ComputeUserMap이 true일 때만 유효합니다.

Returns

원시 깊이 데이터 배열

3.10.2.6 GetUserPosition()

```
Vector3 KinectManager.GetUserPosition (
    uint UserId)
```

GetUserPosition 메소드는 주어진 사용자의 위치를 반환합니다.

Parameters

<i>UserId</i>	사용자의 ID
---------------	---------

Returns

사용자의 위치(Vector3)

3.10.2.7 IsPlayerCalibrated()

```
bool KinectManager.IsPlayerCalibrated (
    uint UserId)
```

IsPlayerCalibrated 메소드는 주어진 사용자가 보정되었는지 확인합니다.

Parameters

<i>User↔ Id</i>	사용자의 ID
---------------------	---------

Returns

사용자가 보정되었으면 true, 그렇지 않으면 false

3.10.2.8 IsUserDetected()

```
bool KinectManager.IsUserDetected ()
```

IsUserDetected 메소드는 최소한 하나의 사용자가 감지되었는지 확인합니다.

Returns

사용자가 감지되었으면 true, 그렇지 않으면 false

The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectManager.cs

3.11 KinectWrapper Class Reference

[KinectWrapper](#) 클래스는 Kinect 센서와 상호작용하기 위한 메소드와 구조체를 포함합니다.

Classes

- struct [ColorBuffer](#)
- struct [ColorCust](#)
- class **Constants**
- struct [DepthBuffer](#)
- interface [INuiFrameTexture](#)
- class [NuiImageBuffer](#)
- struct [NuiImageFrame](#)
- struct [NuiImageViewArea](#)
- struct [NuiLockedRect](#)
- struct [NuiSkeletonBoneOrientation](#)
- struct [NuiSkeletonBoneRotation](#)
- struct [NuiSkeletonData](#)
- struct [NuiSkeletonFrame](#)
- struct [NuiSurfaceDesc](#)
- struct [NuiTransformSmoothParameters](#)

Public Types

- enum **NuiInitializeFlags** : uint {
UsesAudio = 0x10000000 , **UsesDepthAndPlayerIndex** = 0x00000001 , **UsesColor** = 0x00000002 ,
UsesSkeleton = 0x00000008 ,
UsesDepth = 0x00000020 , **UsesHighQualityColor** = 0x00000040 }
- enum **NuiErrorCodes** : uint {
FrameNoData = 0x83010001 , **StreamNotEnabled** = 0x83010002 , **ImageStreamInUse** = 0x83010003 ,
FrameLimitExceeded = 0x83010004 ,
FeatureNotInitialized = 0x83010005 , **DeviceNotGenuine** = 0x83010006 , **InsufficientBandwidth** =
0x83010007 , **DeviceNotSupported** = 0x83010008 ,
DeviceInUse = 0x83010009 , **DatabaseNotFound** = 0x8301000D , **DatabaseVersionMismatch** =
0x8301000E , **HardwareFeatureUnavailable** = 0x8301000F ,
DeviceNotConnected = 0x83010014 , **DeviceNotReady** = 0x83010015 , **SkeletalEngineBusy** =
0x830100AA , **DeviceNotPowered** = 0x8301027F }
- enum **NuiSkeletonPositionIndex** : int {
HipCenter = 0 , **Spine** = 1 , **ShoulderCenter** = 2 , **Head** = 3 ,
ShoulderLeft = 4 , **ElbowLeft** = 5 , **WristLeft** = 6 , **HandLeft** = 7 ,
ShoulderRight = 8 , **ElbowRight** = 9 , **WristRight** = 10 , **HandRight** = 11 ,
HipLeft = 12 , **KneeLeft** = 13 , **AnkleLeft** = 14 , **FootLeft** = 15 ,
HipRight = 16 , **KneeRight** = 17 , **AnkleRight** = 18 , **FootRight** = 19 ,
Count = 20 }
- enum **NuiSkeletonPositionTrackingState** { **NotTracked** = 0 , **Inferred** , **Tracked** }
- enum **NuiSkeletonTrackingState** { **NotTracked** = 0 , **PositionOnly** , **SkeletonTracked** }
- enum **NuiImageType** {
DepthAndPlayerIndex = 0 , **Color** , **ColorYUV** , **ColorRawYUV** ,
Depth }
- enum **NuiImageResolution** {
resolutionInvalid = -1 , **resolution80x60** = 0 , **resolution320x240** = 1 , **resolution640x480** = 2 ,
resolution1280x960 = 3 }
- enum **NuiImageStreamFlags** { **None** = 0x00000000 , **SupressNoFrameData** = 0x0001000 , **EnableNear↵**
Mode = 0x00020000 , **TooFarIsNonZero** = 0x0004000 }
- enum **FrameEdges** {
None = 0 , **Right** = 1 , **Left** = 2 , **Top** = 4 ,
Bottom = 8 }

Public Member Functions

- static int **NuiInitialize** (NuiInitializeFlags dwFlags)
- static void **NuiShutdown** ()
- static int **NuiCameraElevationSetAngle** (int angle)
- static int **NuiCameraElevationGetAngle** (out int pAngleDegrees)
- static int **NuiImageGetColorPixelCoordinatesFromDepthPixelAtResolution** (NuiImageResolution e↵
ColorResolution, NuiImageResolution eDepthResolution, ref [NuiImageViewArea](#) pcViewArea, int lDepthX,
int lDepthY, ushort sDepthValue, out int plColorX, out int plColorY)
- static int **NuiGetSensorCount** (out int pCount)
- static int **NuiSkeletonTrackingEnable** (IntPtr hNextFrameEvent, uint dwFlags)
- static int **NuiSkeletonGetNextFrame** (uint dwMillisecondsToWait, ref [NuiSkeletonFrame](#) pSkeletonFrame)
- static int **NuiTransformSmooth** (ref [NuiSkeletonFrame](#) pSkeletonFrame, ref [NuiTransformSmoothParameters](#)
pSmoothingParams)
- static int **NuiSkeletonCalculateBoneOrientations** (ref [NuiSkeletonData](#) pSkeletonData, [NuiSkeletonBoneOrientation](#)[]
pBoneOrientations)
- static int **NuiImageStreamOpen** (NuiImageType eImageType, NuiImageResolution eResolution, uint dw↵
ImageFrameFlags_NotUsed, uint dwFrameLimit, IntPtr hNextFrameEvent, ref IntPtr phStreamHandle)
- static int **NuiImageStreamGetNextFrame** (IntPtr phStreamHandle, uint dwMillisecondsToWait, ref IntPtr
ppclImageFrame)

- static int **NuImageStreamReleaseFrame** (IntPtr phStreamHandle, IntPtr ppclImageFrame)
- static int **NuImageStreamSetImageFrameFlags** (IntPtr phStreamHandle, NuImageStreamFlags dv↔ ImageFrameFlags)
- static int **NuImageResolutionToSize** (NuImageResolution eResolution, out uint frameWidth, out uint frameHeight)

Static Public Member Functions

- static string **GetNuiErrorString** (int hr)
Kinect 오류 코드를 문자열로 변환합니다.
- static int **GetDepthWidth** ()
- static int **GetDepthHeight** ()
- static int **GetColorWidth** ()
- static int **GetColorHeight** ()
- static Vector3 **MapSkeletonPointToDepthPoint** (Vector3 skeletonPoint)
스켈레톤 포인트를 깊이 포인트로 매핑합니다.
- static int **GetSkeletonJointParent** (int jointIndex)
주어진 조인트 인덱스의 부모 조인트 인덱스를 반환합니다.
- static int **GetSkeletonMirroredJoint** (int jointIndex)
주어진 조인트 인덱스의 반대편 조인트 인덱스를 반환합니다.
- static bool **PollSkeleton** (ref NuiTransformSmoothParameters smoothParameters, ref NuiSkeletonFrame skeletonFrame)
스켈레톤 프레임의 새로운 데이터를 가져오고 스무딩을 적용합니다.
- static bool **PollColor** (IntPtr colorStreamHandle, ref byte[] videoBuffer, ref Color32[] colorImage)
컬러 이미지 프레임을 가져옵니다.
- static bool **PollDepth** (IntPtr depthStreamHandle, bool isNearMode, ref ushort[] depthPlayerData)
깊이 이미지 프레임을 가져옵니다.
- static void **GetSkeletonJointOrientation** (ref Vector3[] jointsPos, ref bool[] jointsTracked, ref Matrix4x4[] jointOrients)
조인트 위치와 추적 상태에 따라 조인트 방향을 계산합니다.

3.11.1 Detailed Description

[KinectWrapper](#) 클래스는 Kinect 센서와 상호작용하기 위한 메소드와 구조체를 포함합니다.

3.11.2 Member Function Documentation

3.11.2.1 GetNuiErrorString()

```
static string KinectWrapper.GetNuiErrorString (
    int hr) [static]
```

Kinect 오류 코드를 문자열로 변환합니다.

Parameters

<i>hr</i>	오류 코드
-----------	-------

Returns

오류 메시지

3.11.2.2 GetSkeletonJointOrientation()

```
static void KinectWrapper.GetSkeletonJointOrientation (
    ref Vector3[] jointsPos,
    ref bool[] jointsTracked,
    ref Matrix4x4[] jointOrientations) [static]
```

조인트 위치와 추적 상태에 따라 조인트 방향을 계산합니다.

Parameters

<i>jointsPos</i>	조인트 위치 배열
<i>jointsTracked</i>	조인트 추적 상태 배열
<i>jointOrientations</i>	조인트 방향 배열

3.11.2.3 GetSkeletonJointParent()

```
static int KinectWrapper.GetSkeletonJointParent (
    int jointIndex) [static]
```

주어진 조인트 인덱스의 부모 조인트 인덱스를 반환합니다.

Parameters

<i>jointIndex</i>	조인트 인덱스
-------------------	---------

Returns

부모 조인트 인덱스

3.11.2.4 GetSkeletonMirroredJoint()

```
static int KinectWrapper.GetSkeletonMirroredJoint (
    int jointIndex) [static]
```

주어진 조인트 인덱스의 반대편 조인트 인덱스를 반환합니다.

Parameters

<i>jointIndex</i>	조인트 인덱스
-------------------	---------

Returns

반대편 조인트 인덱스

3.11.2.5 MapSkeletonPointToDepthPoint()

```
static Vector3 KinectWrapper.MapSkeletonPointToDepthPoint (
    Vector3 skeletonPoint) [static]
```

스켈레톤 포인트를 깊이 포인트로 매핑합니다.

Parameters

<i>skeletonPoint</i>	스켈레톤 포인트
----------------------	----------

Returns

매핑된 깊이 포인트

3.11.2.6 PollColor()

```
static bool KinectWrapper.PollColor (
    IntPtr colorStreamHandle,
    ref byte[] videoBuffer,
    ref Color32[] colorImage) [static]
```

컬러 이미지 프레임을 가져옵니다.

Parameters

<i>colorStreamHandle</i>	컬러 스트림 핸들
<i>videoBuffer</i>	비디오 버퍼
<i>colorImage</i>	컬러 이미지 데이터

Returns

새로운 컬러 데이터의 여부

3.11.2.7 PollDepth()

```
static bool KinectWrapper.PollDepth (
    IntPtr depthStreamHandle,
    bool isNearMode,
    ref ushort[] depthPlayerData) [static]
```

깊이 이미지 프레임을 가져옵니다.

Parameters

<i>depthStreamHandle</i>	깊이 스트림 핸들
<i>isNearMode</i>	근거리 모드 여부
<i>depthPlayerData</i>	깊이 및 플레이어 데이터

Returns

새로운 깊이 데이터의 여부

3.11.2.8 PollSkeleton()

```
static bool KinectWrapper.PollSkeleton (
    ref NuiTransformSmoothParameters smoothParameters,
    ref NuiSkeletonFrame skeletonFrame) [static]
```

스켈레톤 프레임의 새로운 데이터를 가져오고 스무딩을 적용합니다.

Parameters

<i>smoothParameters</i>	스무딩 매개 변수
<i>skeletonFrame</i>	스켈레톤 프레임

Returns

새로운 스켈레톤 데이터의 여부

The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.12 KinectWrapper.NuImageBuffer Class Reference

Public Attributes

- int **m_Width**
- int **m_Height**
- int **m_BytesPerPixel**
- IntPtr **m_pBuffer**

The documentation for this class was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.13 KinectWrapper.NuImageFrame Struct Reference

Public Attributes

- Int64 **liTimeStamp**
- uint **dwFrameNumber**
- NuImageType **eImageType**
- NuImageResolution **eResolution**
- IntPtr **pFrameTexture**
- uint **dwFrameFlags_Used**
- [NuImageViewArea](#) **ViewArea_Used**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.14 KinectWrapper.NuiImageAreaViewArea Struct Reference

Public Attributes

- int **eDigitalZoom**
- int **ICenterX**
- int **ICenterY**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.15 KinectWrapper.NuiLockedRect Struct Reference

Public Attributes

- int **pitch**
- int **size**
- IntPtr **pBits**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.16 KinectWrapper.NuiSkeletonBoneOrientation Struct Reference

Public Attributes

- NuiSkeletonPositionIndex **endJoint**
- NuiSkeletonPositionIndex **startJoint**
- [NuiSkeletonBoneRotation](#) **hierarchicalRotation**
- [NuiSkeletonBoneRotation](#) **absoluteRotation**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.17 KinectWrapper.NuiSkeletonBoneRotation Struct Reference

Public Attributes

- Matrix4x4 **rotationMatrix**
- Quaternion **rotationQuaternion**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.18 KinectWrapper.NuiSkeletonData Struct Reference

Public Attributes

- NuiSkeletonTrackingState **eTrackingState**
- uint **dwTrackingID**
- uint **dwEnrollmentIndex_NotUsed**
- uint **dwUserIndex**
- Vector4 **Position**
- Vector4[] **SkeletonPositions**
- NuiSkeletonPositionTrackingState[] **eSkeletonPositionTrackingState**
- uint **dwQualityFlags**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.19 KinectWrapper.NuiSkeletonFrame Struct Reference

Public Attributes

- Int64 **liTimeStamp**
- uint **dwFrameNumber**
- uint **dwFlags**
- Vector4 **vFloorClipPlane**
- Vector4 **vNormalToGravity**
- [NuiSkeletonData\[\]](#) **SkeletonData**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.20 KinectWrapper.NuiSurfaceDesc Struct Reference

Public Attributes

- uint **width**
- uint **height**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

3.21 KinectWrapper.NuiTransformSmoothParameters Struct Reference

Public Attributes

- float **fSmoothing**
- float **fCorrection**
- float **fPrediction**
- float **fJitterRadius**
- float **fMaxDeviationRadius**

The documentation for this struct was generated from the following file:

- C:/Users/NICEDNB/Desktop/test/KinectWrapper.cs

Index

- AvatarController, [5](#)
 - boneIndex2JointMap, [9](#)
 - boneIndex2MirrorJointMap, [9](#)
 - Kinect2AvatarPos, [6](#)
 - Kinect2AvatarRot, [7](#)
 - MapBones, [7](#)
 - MoveAvatar, [7](#)
 - specIndex2JointMap, [9](#)
 - specIndex2MirrorJointMap, [10](#)
 - SuccessfulCalibration, [7](#)
 - TransformBone, [8](#)
 - TransformSpecialBone, [8](#)
 - UpdateAvatar, [8](#)
- AvatarControllerClassic, [10](#)
 - MapBones, [12](#)
- boneIndex2JointMap
 - AvatarController, [9](#)
- boneIndex2MirrorJointMap
 - AvatarController, [9](#)
- CheckForGesture
 - Kinect Gestures, [17](#)
- DetectGesture
 - KinectManager, [20](#)
- GestureCancelled
 - Kinect Gestures.GestureListenerInterface, [14](#)
- GestureCompleted
 - Kinect Gestures.GestureListenerInterface, [15](#)
- GestureInProgress
 - Kinect Gestures.GestureListenerInterface, [15](#)
- GetDepthForPixel
 - KinectManager, [20](#)
- GetNuiErrorString
 - KinectWrapper, [24](#)
- GetPlayer1ID
 - KinectManager, [20](#)
- GetPlayer2ID
 - KinectManager, [21](#)
- GetRawDepthMap
 - KinectManager, [21](#)
- GetSkeletonJointOrientation
 - KinectWrapper, [24](#)
- GetSkeletonJointParent
 - KinectWrapper, [25](#)
- GetSkeletonMirroredJoint
 - KinectWrapper, [25](#)
- GetPosition
 - KinectManager, [21](#)
- IsPlayerCalibrated
 - KinectManager, [21](#)
- IsUserDetected
 - KinectManager, [22](#)
- Kinect2AvatarPos
 - AvatarController, [6](#)
- Kinect2AvatarRot
 - AvatarController, [7](#)
- Kinect Gestures, [17](#)
 - CheckForGesture, [17](#)
- Kinect Gestures.GestureData, [13](#)
- Kinect Gestures.GestureListenerInterface, [14](#)
 - GestureCancelled, [14](#)
 - GestureCompleted, [15](#)
 - GestureInProgress, [15](#)
 - UserDetected, [16](#)
 - UserLost, [16](#)
- KinectManager, [18](#)
 - DetectGesture, [20](#)
 - GetDepthForPixel, [20](#)
 - GetPlayer1ID, [20](#)
 - GetPlayer2ID, [21](#)
 - GetRawDepthMap, [21](#)
 - GetPosition, [21](#)
 - IsPlayerCalibrated, [21](#)
 - IsUserDetected, [22](#)
- KinectWrapper, [22](#)
 - GetNuiErrorString, [24](#)
 - GetSkeletonJointOrientation, [24](#)
 - GetSkeletonJointParent, [25](#)
 - GetSkeletonMirroredJoint, [25](#)
 - MapSkeletonPointToDepthPoint, [25](#)
 - PolIColor, [26](#)
 - PolIDepth, [26](#)
 - PolISkeleton, [26](#)
- KinectWrapper.ColorBuffer, [13](#)
- KinectWrapper.ColorCust, [13](#)
- KinectWrapper.DepthBuffer, [13](#)
- KinectWrapper.INuiFrameTexture, [16](#)
- KinectWrapper.NuiImageBuffer, [27](#)
- KinectWrapper.NuiImageFrame, [27](#)
- KinectWrapper.NuiImageViewArea, [28](#)
- KinectWrapper.NuiLockedRect, [28](#)
- KinectWrapper.NuiSkeletonBoneOrientation, [28](#)
- KinectWrapper.NuiSkeletonBoneRotation, [28](#)
- KinectWrapper.NuiSkeletonData, [29](#)
- KinectWrapper.NuiSkeletonFrame, [29](#)

KinectWrapper.NuiSurfaceDesc, [29](#)
KinectWrapper.NuiTransformSmoothParameters, [29](#)

MapBones
 AvatarController, [7](#)
 AvatarControllerClassic, [12](#)
MapSkeletonPointToDepthPoint
 KinectWrapper, [25](#)
MoveAvatar
 AvatarController, [7](#)

PollColor
 KinectWrapper, [26](#)
PollDepth
 KinectWrapper, [26](#)
PollSkeleton
 KinectWrapper, [26](#)

specIndex2JointMap
 AvatarController, [9](#)
specIndex2MirrorJointMap
 AvatarController, [10](#)
SuccessfulCalibration
 AvatarController, [7](#)

TransformBone
 AvatarController, [8](#)
TransformSpecialBone
 AvatarController, [8](#)

UpdateAvatar
 AvatarController, [8](#)
UserDetected
 KinectGestures.GestureListenerInterface, [16](#)
UserLost
 KinectGestures.GestureListenerInterface, [16](#)