Magic Mirror

Generated by Doxygen 1.8.16

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 AbductionMove Class Reference

Inheritance diagram for AbductionMove:

class$_{a}bduction_{m}ove$

Public Member Functions

- bool **AreRepetitionsDone** ()

  Check if the exercise has been performed a given amount of times.

Public Attributes

- GameObject **movingMarker**
- GameObject **pivotMarker**
- Vector3 **pivotJointPosition**
- float **minAngleInDegrees**
- float **maxAngleInDegrees**
- float **radius** = 1f
- float **speed** = 1f
- int **numRepetitions** = 3

3.1.1 Detailed Description

Brief Abduction/Adduction movement represented in 2D.

3.1.2 Member Function Documentation

3.1.2.1 AreRepetitionsDone()

```
bool AbductionMove.AreRepetitionsDone ( )
```

Check if the exercise has been performed a given amount of times.

Returns

True when all repetitions are completed, False otherwise.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/AbductionMove.cs

3.2 ArmRoutine Class Reference

Inheritance diagram for ArmRoutine:

class$_{a}rm_{r}outine$

Public Attributes

- GameObject **bodySensor**
- GameObject **abductionObject**

3.2.1 Member Data Documentation

3.2.1.1 bodySensor

```
GameObject ArmRoutine.bodySensor
```

Reference to body sensor.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/ArmRoutine.cs

3.3 BodyAnalysis Class Reference

Static Public Member Functions

- static float **ComputeBodyHeight** (Body body)

  Compute the height of the currently tracked body in 3D space, for scaling purposes. Methodology obtained from `https://pterneas.com/kinect/`.
- static UnityEngine.Vector3[ ] **convertToUnityPosition** (Dictionary< JointType, Windows.Kinect.Joint > kinectJoint)

  Convert Kinect's 3D position to Unity's Vector3 Kinect reads and stores body joints using its own "Joint" structure. For displaying and further analysis purposes, we need to convert to Unity's Vector3 structure.
- static UnityEngine.Quaternion[ ] **convertToUnityOrientation** (Dictionary< JointType, JointOrientation > kinectJoint)

  Convert Kinect's orientation to Unity's Quaternion Kinect has its own Quaternion representation for body joint's orientation. This function converts that representation to Unity's Quaternion for further analysis.
- static float **ComputeArmLength** (Body body, bool rightArm)

  Compute arm length in 3D space.
- static float **ComputeLegLength** (Body body, bool rightLeg)

  Compute leg length in 3D scene.
- static float **FlexionSpineAngle** (Body body, bool coronalPlane)

  Angle of the spine in flexion movement.
- static float **AbductionShoulderAngle** (Body body, bool rightArm)

  Angle of the shoulder in abduction movement.
- static float **AbductionLegAngle** (Body body, bool rightLeg)

  Angle of the leg in abduction movement (taking hip as rotation pivot).
- static float **FlexionShoulderAngle** (Body body, bool rightArm)

  Angle of shoulder in Flexion movement.
- static float **FlexionElbowAngle** (Body body, bool rightArm)

  Angle of elbow in flexion movement.
- static float **FlexionLegAngle** (Body body, bool rightLeg)

  Angle of leg in Flexion movement.
- static float **FlexionKneeAngle** (Body body, bool rightLeg)

  Angle of knee in flexion movement.
- static void **SetMapper** (CoordinateMapper mapper)

  Set Kinect's CoordinateMapper for conversion from 3D to screen coordinates.
- static float **ComputeBodyHeightInScreenCoordinates** (Body body, UnityEngine.Vector2 scale, Unity↩ Engine.Vector2 position)

  Compute the height of the currently tracked body in 2D space, for scaling purposes Methodology obtained from `https://pterneas.com/kinect/`.
- static UnityEngine.Vector2[ ] **convertToUnity2DPosition** (Dictionary< JointType, Windows.Kinect.Joint > kinectJoint, UnityEngine.Vector2 scale, UnityEngine.Vector2 position)
- static float **ComputeArmLength2D** (Body body, UnityEngine.Vector2 scale, UnityEngine.Vector2 position, bool rightArm)

  Compute arm length in screen coordinates.
- static float **ComputeLegLength2D** (Body body, UnityEngine.Vector2 scale, UnityEngine.Vector2 position, bool rightLeg)

  Compute leg length in screen coordinates.

3.3.1 Member Function Documentation

3.3.1.1 AbductionLegAngle()

```
static float BodyAnalysis.AbductionLegAngle (
            Body body,
            bool rightLeg )  [static]
```

Angle of the leg in abduction movement (taking hip as rotation pivot).

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightLeg	true for right leg, false for left leg.

Returns

Angle of leg in abduction movement.

3.3.1.2 AbductionShoulderAngle()

```
static float BodyAnalysis.AbductionShoulderAngle (
            Body body,
            bool rightArm )  [static]
```

Angle of the shoulder in abduction movement.

Parameters

Reference	to a body detected and tracked by the kinect sensor.
rightArm	true for right arm, false for left arm.

Returns

Angle of shoulder in abduction movement.

3.3.1.3 ComputeArmLength()

```
static float BodyAnalysis.ComputeArmLength (
            Body body,
            bool rightArm )  [static]
```

Compute arm length in 3D space.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightArm	True for right arm, False for left arm.

Returns

Arm length in 3D space.

3.3.1.4 ComputeArmLength2D()

```
static float BodyAnalysis.ComputeArmLength2D (
             Body body,
             UnityEngine.Vector2 scale,
             UnityEngine.Vector2 position,
             bool rightArm )  [static]
```

Compute arm length in screen coordinates.

Parameters

kinectJoint	Position of the body joints in Kinect's Joint structure.
scale	Size of the screen where the model will be placed.
position	Position of the screen where the model will be placed.
rightArm	True for right arm, False for left arm.

Returns

Arm length in screen coordinates.

3.3.1.5 ComputeBodyHeight()

```
static float BodyAnalysis.ComputeBodyHeight (
             Body body )  [static]
```

Compute the height of the currently tracked body in 3D space, for scaling purposes. Methodology obtained from `https://pterneas.com/kinect/`.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
scale	The current size of the "screen" where the 3D model will be drawn.

Returns

Height of the tracked body in 3D space.

3.3.1.6 ComputeBodyHeightInScreenCoordinates()

```
static float BodyAnalysis.ComputeBodyHeightInScreenCoordinates (
            Body body,
            UnityEngine.Vector2 scale,
            UnityEngine.Vector2 position )  [static]
```

Compute the height of the currently tracked body in 2D space, for scaling purposes Methodology obtained from `https://pterneas.com/kinect/`.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
scale	The size of the 2D screen where the model will be drawn.
position	The position where the screen is placed. Relevant for the conversion between Kinect's Vector2 and Unity's Vector2.

Returns

Height of the tracked body in 2D coordinates.

3.3.1.7 ComputeLegLength()

```
static float BodyAnalysis.ComputeLegLength (
            Body body,
            bool rightLeg )  [static]
```

Compute leg length in 3D scene.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightLeg	True for right leg, False for left leg.

Returns

Leg length in 3D space.

3.3.1.8 ComputeLegLength2D()

```
static float BodyAnalysis.ComputeLegLength2D (
            Body body,
            UnityEngine.Vector2 scale,
            UnityEngine.Vector2 position,
            bool rightLeg )  [static]
```

Compute leg length in screen coordinates.

Parameters

kinectJoint	Position of the body joints in Kinect's Joint structure.
scale	Size of the screen where the model will be placed.
position	Position of the screen where the model will be placed.
rightLeg	True for right leg, False for left leg.

Returns

Leg length in screen coordinates.

3.3.1.9 convertToUnity2DPosition()

```
static UnityEngine.Vector2 [] BodyAnalysis.convertToUnity2DPosition (
            Dictionary< JointType, Windows.Kinect.Joint > kinectJoint,
            UnityEngine.Vector2 scale,
            UnityEngine.Vector2 position )  [static]
```

Convert Kinect's 2D position to Unity's Vector2 Kinect reads and stores body joints using its own "Joint" structure. For displaying and further 2D analysis purposes, we need to convert to Unity's Vector2 structure.

Parameters

kinectJoint	Position of the body joints in Kinect's Joint structure.
scale	Size of the screen where the model will be placed.
position	Position of the screen where the model will be placed.

Returns

Position of the body joints in Unity's Vector2 structure, more suitable for further analysis.

3.3.1.10 convertToUnityOrientation()

```
static UnityEngine.Quaternion [] BodyAnalysis.convertToUnityOrientation (
            Dictionary< JointType, JointOrientation > kinectJoint )  [static]
```

Convert Kinect's orientation to Unity's Quaternion Kinect has its own Quaternion representation for body joint's orientation. This function converts that representation to Unity's Quaternion for further analysis.

Parameters

kinectJoint	Position of thbody joints in Kinect's Joint structure.

Returns

Vector of Unity's Quaternion representing each body joint's orientation.

3.3.1.11 convertToUnityPosition()

```
static UnityEngine.Vector3 [] BodyAnalysis.convertToUnityPosition (
            Dictionary< JointType, Windows.Kinect.Joint > kinectJoint )  [static]
```

Convert Kinect's 3D position to Unity's Vector3 Kinect reads and stores body joints using its own "Joint" structure. For displaying and further analysis purposes, we need to convert to Unity's Vector3 structure.

Parameters

kinectJoint	Position of the body joints in Kinect's Joint structure.
scale	The size of the screen where the model will be placed.

Returns

Position of the body joints in Unity's Vector3 structure, more suitable for further analysis.

3.3.1.12 FlexionElbowAngle()

```
static float BodyAnalysis.FlexionElbowAngle (
            Body body,
            bool rightArm )  [static]
```

Angle of elbow in flexion movement.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightArm	true for right arm, false for left arm.

Returns

angle of elbow in flexion movement.

3.3.1.13 FlexionKneeAngle()

```
static float BodyAnalysis.FlexionKneeAngle (
            Body body,
            bool rightLeg )  [static]
```

Angle of knee in flexion movement.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightLeg	true for right leg, false for left leg.

Returns

Angle of knee in flexion movement.

3.3.1.14 FlexionLegAngle()

```
static float BodyAnalysis.FlexionLegAngle (
             Body body,
             bool rightLeg )  [static]
```

Angle of leg in Flexion movement.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightLeg	true for right leg, false for left leg.

Returns

Angle of leg in flexion movement.

3.3.1.15 FlexionShoulderAngle()

```
static float BodyAnalysis.FlexionShoulderAngle (
             Body body,
             bool rightArm )  [static]
```

Angle of shoulder in Flexion movement.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
rightArm	true for right arm, false for left arm.

Returns

Angle of shoulder in flexion movement.

3.3.1.16 FlexionSpineAngle()

```
static float BodyAnalysis.FlexionSpineAngle (
            Body body,
            bool coronalPlane )  [static]
```

Angle of the spine in flexion movement.

Parameters

body	Reference to a body detected and tracked by the kinect sensor.
coronalPlane	true for coronal plane, false for sagital plane.

Returns

Angle of spine in flexion movement.

3.3.1.17 SetMapper()

```
static void BodyAnalysis.SetMapper (
            CoordinateMapper mapper )  [static]
```

Set Kinect's CoordinateMapper for conversion from 3D to screen coordinates.

Parameters

mapper	Reference to a CoordinateMapper.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/BodyAnalysis.cs

3.4 BoxSkeleton Class Reference

Inheritance diagram for BoxSkeleton:

class$_box_skeleton$

Public Attributes

- GameObject **BodySourceManager**
- GameObject **Marker**
- float **scale** = 1.0f

3.4.1 Detailed Description

Brief Display the body joints as boxes or any given marker (sphere, capsule, etc.).

This script is part of the "Simple Body" scene.

3.4.2 Member Data Documentation

3.4.2.1 BodySourceManager

```
GameObject BoxSkeleton.BodySourceManager
```

Reference to **SensorBody** (p. 17).

3.4.2.2 Marker

```
GameObject BoxSkeleton.Marker
```

Reference to the solid used as marker (a box for instance).

3.4.2.3 scale

```
float BoxSkeleton.scale = 1.0f
```

The scale at which the marker will be drawn on each body joint.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/BoxSkeleton.cs

3.5 DisplayRGB Class Reference

Inheritance diagram for DisplayRGB:

class$_{d}isplay_{rgb}$

Public Attributes

- GameObject **sensorRGB**

3.5.1 Detailed Description

Brief Display Kinect v2 RGB image on a solid.

Attach this script to a cube with aspect ratio of 16:9 (same as RGB image retrieved by the sensor) in one of their faces. Drag the object containing the **SensorRGB** (p. 18) script to the cube containing this script.

3.5.2 Member Data Documentation

3.5.2.1 sensorRGB

```
GameObject DisplayRGB.sensorRGB
```

Game object containing a **SensorRGB** (p. 18) script.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/DisplayRGB.cs

3.6 HandTracker Class Reference

Inheritance diagram for HandTracker:

class$_{h}and_{t}racker$

Public Attributes

- GameObject **bodySensor**
- GameObject **marker**
- bool **trackRightHand** = true
- bool **trackLeftHand** = false

3.6.1 Detailed Description

Brief Attach this class to the body controller. This is useful for projects that include a hand tracker.

This script is part of the "Simple hand tracking" scene.

3.6.2 Member Data Documentation

3.6.2.1 bodySensor

```
GameObject HandTracker.bodySensor
```

Reference to body sensor.

3.6.2.2 marker

```
GameObject HandTracker.marker
```

Reference to the solid used as marker (a box for instance).

3.6.2.3 trackLeftHand

```
bool HandTracker.trackLeftHand = false
```

True if tracking left hand.

3.6.2.4 trackRightHand

```
bool HandTracker.trackRightHand = true
```

True if tracking right hand.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/HandTracker.cs

3.7 SensorBody Class Reference

Inheritance diagram for SensorBody:

class$_s ensor_b ody$

Public Member Functions

- Body[ ] **GetBodies** ()
  Get a vector to all bodies read by the sensor.
- CoordinateMapper **GetMapper** ()
  Get a mapper to convert between Kinect's 3D and 2D spaces.

3.7.1 Detailed Description

Brief Reads body joints from Kinect v2 sensor.

The Kinect v2 sensor can read and track up to six bodies.

Attach this script to an empty object. Make a reference from this object to the one controlling the display of the body.

3.7.2 Member Function Documentation

3.7.2.1 GetBodies()

```
Body [] SensorBody.GetBodies ( )
```

Get a vector to all bodies read by the sensor.

Returns

Vector to all bodies read by the sensor. The class Body is native from Windows.Kinect.

3.7.2.2 GetMapper()

```
CoordinateMapper SensorBody.GetMapper ( )
```

Get a mapper to convert between Kinect's 3D and 2D spaces.

Returns

Coordinate mapper to transform between 3D and 2D spaces.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/SensorBody.cs

3.8 SensorRGB Class Reference

Inheritance diagram for SensorRGB:

class$_sensor_{rgb}$

Public Member Functions

- Texture2D **GetColorTexture** ()
  Get a texture containing the last frame acquired by the Kinect v2 RGB sensor.

Properties

- int **ColorWidth** `[get]`
- int **ColorHeight** `[get]`

3.8.1 Detailed Description

Brief Connect to sensor Kinect v2 and retrieve RGB color frame as texture to be applied on solid.

Connect this script to an empty object, then import that empty object to the object containing the **DisplayRGB** (p. 15) script.

This script was obtained from the Kinect v2 for Unity documentation.

3.8.2 Member Function Documentation

3.8.2.1 GetColorTexture()

```
Texture2D SensorRGB.GetColorTexture ( )
```

Get a texture containing the last frame acquired by the Kinect v2 RGB sensor.

Apply this texture to a solid body, for example, a cube. The size of two of the faces of the cube should have the aspect ratio of 16:9, the same aspect ratio of the Kinect v2 sensor.

Returns

Texture containing the last frame acquired by the sensor.

3.8.3 Property Documentation

3.8.3.1 ColorHeight

```
int SensorRGB.ColorHeight  [get]
```

Get image height.

3.8.3.2 ColorWidth

```
int SensorRGB.ColorWidth  [get]
```

Get image width.

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/SensorRGB.cs

3.9 SimpleBodyTracker Class Reference

Inheritance diagram for SimpleBodyTracker:

class$_{s}imple_{b}ody_{t}racker$

Public Attributes

- GameObject **bodySensor**
- GameObject **marker**

3.9.1 Detailed Description

Brief Display the body joins as boxes (or any given marker) in front of the RGB image.

This script is part of the "Body and RGB" scene.

3.9.2 Member Data Documentation

3.9.2.1 bodySensor

```
GameObject SimpleBodyTracker.bodySensor
```

Reference to body sensor.

3.9.2.2 marker

```
GameObject SimpleBodyTracker.marker
```

Reference to the solid used as marker (a box for instance).

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/SimpleBodyTracker.cs

3.10 ViewBodyAngles Class Reference

Inheritance diagram for ViewBodyAngles:

class$_{v}iew_{b}ody_{a}ngles$

Public Attributes

- GameObject **bodySensor**
- GameObject **marker**
- GameObject **textBox**

3.10.1 Member Data Documentation

3.10.1.1 bodySensor

```
GameObject ViewBodyAngles.bodySensor
```

Reference to body sensor.

3.10.1.2 marker

```
GameObject ViewBodyAngles.marker
```

Reference to the solid used as marker (a box for instance).

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

- D:/DocsSheila/Unity/Magic Mirror/Assets/Scripts/ViewBodyAngles.cs

Index