

# Table of Contents

- Uralstech.UXR.Utilities ..... 2
  - SimpleCameraFollower ..... 3
  - SimpleMovementController ..... 6
  - TextInputField ..... 8
  - TextInputFieldVoiceHandler ..... 13
  - XRKeyboardManager ..... 16

# Namespace Uralstech.UXR.Utilities

## Classes

### [SimpleCameraFollower](#)

Simple follow script which allows the follower to be offset from the player.

### [SimpleMovementController](#)

Simple movement controller using Meta Quest input.

### [TextInputField](#)

Text input field compatible with OVRVirtualKeyboard input.

### [TextInputFieldVoiceHandler](#)

Sister-script to [TextInputField](#) which adds support for voice typing through Meta's Voice SDK.

### [XRKeyboardManager](#)

Wrapper class for Meta's Virtual Keyboard.

# Class SimpleCameraFollower

Namespace: [Uralstech.UXR.Utilities](#)

Simple follow script which allows the follower to be offset from the player.

```
public class SimpleCameraFollower : MonoBehaviour
```

## Inheritance

System.Object ← SimpleCameraFollower

## Fields

### DistanceOffset

Z distance offset from the player.

```
public float DistanceOffset
```

Field Value

float

### HeightOffset

Height (Y) offset from the player.

```
public float HeightOffset
```

Field Value

float

### MovementDelayDistance

Delay before starting to follow player position.

```
public float MovementDelayDistance
```

Field Value

float

## MovementSpeed

Speed at which the object follows the player.

```
public float MovementSpeed
```

Field Value

float

## RotationDelayAngle

Delay before starting to follow player rotation.

```
public float RotationDelayAngle
```

Field Value

float

## RotationOffset

Rotation offset from the player.

```
public Vector3 RotationOffset
```

Field Value

Vector3

## RotationSpeed

Speed at which the object rotates to match the player's head rotation.

```
public float RotationSpeed
```

Field Value

float

## Methods

### LateUpdate()

```
protected void LateUpdate()
```

### Start()

```
protected void Start()
```

# Class SimpleMovementController

Namespace: [Uralstech.UXR.Utilities](#)

Simple movement controller using Meta Quest input.

```
public class SimpleMovementController : DontCreateNewSingleton<SimpleMovementController>
```

## Inheritance

System.Object ← SimpleMovementController

## Fields

### MovementEnabled

Use this boolean to toggle movement.

```
public bool MovementEnabled
```

Field Value

bool

### MovementSpeed

The movement speed.

```
public float MovementSpeed
```

Field Value

float

### RotationSnapAngle

Snap turn angle.

```
public float RotationSnapAngle
```

Field Value

float

## Methods

FixedUpdate()

```
protected void FixedUpdate()
```

Start()

```
protected void Start()
```

# Class TextInputField

Namespace: [Uralstech.UXR.Utilities](#)

Text input field compatible with OVRVirtualKeyboard input.

```
public class TextInputField : Selectable, IPointerClickHandler,  
    OVRVirtualKeyboard.ITextHandler
```

## Inheritance

System.Object ← TextInputField

## Implements

IPointerClickHandler, ITextHandler

## Fields

### OnFieldDeselected

Called when the input field has been deselected.

```
public UnityEvent OnFieldDeselected
```

Field Value

UnityEvent

### OnFieldEdited

Called when the input field has been edited.

```
public UnityEvent<string> OnFieldEdited
```

Field Value

UnityEvent<string>



# OnFieldSelected

Called when the input field has been selected.

```
public UnityEvent OnFieldSelected
```

Field Value

UnityEvent

## Properties

### IsFocused

```
public bool IsFocused { get; }
```

Property Value

bool

### OnTextChanged

```
public Action<string> OnTextChanged { get; set; }
```

Property Value

Action<string>

### SubmitOnEnter

```
public bool SubmitOnEnter { get; set; }
```

Property Value

bool

# Text

The text entered by the user.

```
public string Text { get; }
```

Property Value

string

## onValueChanged

```
[Obsolete("Use TextInputField.OnFieldEdited instead.")]  
public UnityEvent<string> onValueChanged { get; }
```

Property Value

UnityEvent<string>

## text

```
[Obsolete("Use TextInputField.Text instead.")]  
public string text { get; }
```

Property Value

string

## Methods

AppendText(string)

```
public void AppendText(string addedText)
```

## Parameters

**addedText** string

## ApplyBackspace()

```
public void ApplyBackspace()
```

## MoveTextEnd()

```
public void MoveTextEnd()
```

## OnCanvasGroupChanged()

```
protected override void OnCanvasGroupChanged()
```

## OnDestroy()

```
protected override void OnDestroy()
```

## OnDisable()

```
protected override void OnDisable()
```

## OnPointerClick(PointerEventData)

```
public virtual void OnPointerClick(PointerEventData eventData)
```

## Parameters

## SetText(string)

Sets the text in the field.

```
public void SetText(string text)
```

### Parameters

**text** string

The new text.

## SetTextWithoutNotify(string)

Sets the text in the field, without calling [OnFieldEdited](#).

```
public void SetTextWithoutNotify(string text)
```

### Parameters

**text** string

The new text.

## Submit()

```
public void Submit()
```

# Class TextInputFieldVoiceHandler

Namespace: [Uralstech.UXR.Utilities](#)

Sister-script to [TextInputField](#) which adds support for voice typing through Meta's Voice SDK.

```
public class TextInputFieldVoiceHandler : MonoBehaviour
```

## Inheritance

System.Object ← TextInputFieldVoiceHandler

## Remarks

Requires a DictationService somewhere in the scene.

## Fields

### StartRecordingIcon

Optional, icon to set for [ToggleButtonIcon](#) while not recording the user's audio.

```
public Sprite StartRecordingIcon
```

Field Value

Sprite

### StopRecordingIcon

Optional, icon to set for [ToggleButtonIcon](#) while recording the user's audio.

```
public Sprite StopRecordingIcon
```

Field Value

Sprite

# ToggleButton

The button to toggle voice typing.

```
public Button ToggleButton
```

Field Value

Button

# ToggleButtonIcon

Optional, the icon for [ToggleButton](#) that will be changed based on the current recording state.

```
public Image ToggleButtonIcon
```

Field Value

Image

## Properties

### IsRecording

Is the user currently voice typing?

```
public bool IsRecording { get; }
```

Property Value

bool

## Methods

### Start()

```
protected void Start()
```

# Class XRKeyboardManager

Namespace: [Uralstech.UXR.Utilities](#)

Wrapper class for Meta's Virtual Keyboard.

```
public class XRKeyboardManager : Singleton<XRKeyboardManager>
```

## Inheritance

System.Object ← XRKeyboardManager

## Fields

### InitialDistance

The initial distance of the keyboard, when shown, from the player's forward direction.

```
public float InitialDistance
```

Field Value

float

### InitialPosition

The initial position of the keyboard, when shown, relative to the player.

```
public Vector3 InitialPosition
```

Field Value

Vector3

### InitialRotation



The initial rotation of the keyboard, relative to the player's forward direction.

```
public Vector3 InitialRotation
```

Field Value

Vector3

## KeyboardInstance

The current instance of the OVRVirtualKeyboard.

```
public OVRVirtualKeyboard KeyboardInstance
```

Field Value

OVRVirtualKeyboard

## KeyboardPrefab

OVRVirtualKeyboard prefab to spawn when needed.

```
public OVRVirtualKeyboard KeyboardPrefab
```

Field Value

OVRVirtualKeyboard

## LeftControllerDirectTransform

The "direct" transform of the left controller, for example: OVRCameraRig > OVRInteraction > OVRControllers > LeftController > ControllerInteractors > ControllerPokeInteractor > PokeLocation.

```
public Transform LeftControllerDirectTransform
```

## Field Value

Transform

## Remarks

You can add a tracked custom interactor to use as the direct transform:

- Add a new GameObject with a ControllerRef, ControllerPointerPose and ActiveStateTracker under OVRInteraction > OVRControllers > LeftController > ControllerInteractors.
- Drag LeftController into the ControllerRef's "Controller" field.
- Set the "Active State" and "Controller" fields in the ActiveStateTracker and ControllerPointerPose to the ControllerRef.
- Set the Z value in the "Offset" field of the ControllerPointerPose to 0.01.
- (Optional) Enable "Include Children As Dependents" in the ActiveStateTracker and add a sphere as a child of the GameObject.

## LeftControllerRootTransform

The root transform of the left controller, for example: OVRCameraRig > OVRInteraction > OVRControllers > LeftController.

```
public Transform LeftControllerRootTransform
```

## Field Value

Transform

## LeftOVRHand

The OVRHand script for left hand input.

```
public OVRHand LeftOVRHand
```

## Field Value

OVRHand

# NormalHandMaterial

The hand tracking material when the keyboard is not being used.

```
public Material NormalHandMaterial
```

Field Value

Material

# OnKeyboardHidden

Called when the keyboard is destroyed.

```
public UnityEvent OnKeyboardHidden
```

Field Value

UnityEvent

# OnKeyboardShown

Called when the keyboard is created.

```
public UnityEvent OnKeyboardShown
```

Field Value

UnityEvent

# PlayerHead

The player Transform. The keyboard will spawn relative to this object's position. Defaults to first OVRCameraRig found in the scene.

```
public Transform PlayerHead
```

Field Value

Transform

## RightControllerDirectTransform

The "direct" transform of the right controller, for example: OVRCameraRig > OVRInteraction > OVRControllers > RightController > ControllerInteractors > ControllerPokeInteractor > PokeLocation.

```
public Transform RightControllerDirectTransform
```

Field Value

Transform

Remarks

You can add a tracked custom interactor to use as the direct transform:

- Add a new GameObject with a ControllerRef, ControllerPointerPose and ActiveStateTracker under OVRInteraction > OVRControllers > RightController > ControllerInteractors.
- Drag RightController into the ControllerRef's "Controller" field.
- Set the "Active State" and "Controller" fields in the ActiveStateTracker and ControllerPointerPose to the ControllerRef.
- Set the Z value in the "Offset" field of the ControllerPointerPose to 0.01.
- (Optional) Enable "Include Children As Dependents" in the ActiveStateTracker and add a sphere as a child of the GameObject.

## RightControllerRootTransform

The root transform of the right controller, for example: OVRCameraRig > OVRInteraction > OVRControllers > RightController.

```
public Transform RightControllerRootTransform
```

Field Value

Transform

# RightOVRHand

The OVRHand script for right hand input.

```
public OVRHand RightOVRHand
```

Field Value

OVRHand

# TypingHandMaterial

The hand tracking material when the keyboard is being used.

```
public Material TypingHandMaterial
```

Field Value

Material

# UseDifferentMaterialsForHandsWhenTyping

Should the hand tracking material be changed depending on if the keyboard is being used?

```
public bool UseDifferentMaterialsForHandsWhenTyping
```

Field Value

bool

## Properties

### CurrentListener

The object currently bound to the keyboard.

```
public OVRVirtualKeyboard.ITextHandler CurrentListener { get; }
```

Property Value

ITextHandler

## Methods

### RemoveListener(ITextHandler)

Unbinds the given object from the keyboard.

```
public void RemoveListener(OVRVirtualKeyboard.ITextHandler listener)
```

Parameters

**listener** ITextHandler

The object to unbind from the keyboard. Must be the same as [CurrentListener](#).

### SetListener(ITextHandler)

Binds the given object to the keyboard.

```
public void SetListener(OVRVirtualKeyboard.ITextHandler listener)
```

Parameters

**listener** ITextHandler

The object to bind to the keyboard.

### Start()

```
protected void Start()
```