Namespace LastAssets.LastVirtualKeyboard

Classes

VirtualKey

Virtual keyboard key class, assigned when virtualkeyboardlayout creates the keyboard. Mostly used as an animation layer.

VirtualKeyPointer

Pointer class, useful for pointing any raycastable transform to a virtual key

VirtualKeyboard

Main Virtual keyboard class, required for functionality

VirtualKeyboardLayout

Layout component that creates and adjusts the virtual keyboard. This is a required component

VirtualKeyboardLayout.KeyboardLayout

Main components to the keyboard, customization can be done here

VirtualKeyboardLayout.KeyboardRow

Keyboard row, holds a list of KeyboardItems that get aligned horizontally

VirtualKeyboardUtilities

VirtualKeyboard Utility class

Structs

VirtualKeyboardLayout.AdditiveKeyIcon

Additional information to be passed into the creation of a keyboarditem. Can be text or an image

VirtualKeyboardLayout.KeyboardItem

An item that will show up on the keyboard

VirtualKeyboardLayout.VirtualHotkey

Binds an inputsystem action to a key result

Interfaces

IVirtualKeyboardRaycastTarget

Interface for what is raycastable by the virtualkeyboard. It only searches for this interface on hit objects.

Enums

VirtualKeyboardUtilities.UIAnchor

Repesents a Vector2 pivot point used in UnityEngines UI RectTransform

Interface IVirtualKeyboardRaycastTarget

Namespace: [LastAssets](.).[LastVirtualKeyboard](.)

Interface for what is raycastable by the virtualkeyboard. It only searches for this interface on hit objects.

```
public interface IVirtualKeyboardRaycastTarget
```

Methods

GetKey()

```
VirtualKey GetKey()
```

Returns

[VirtualKey](.)

Class VirtualKey

Namespace: [LastAssets](#).[LastVirtualKeyboard](#)

Virtual keyboard key class, assigned when virtualkeyboardlayout creates the keyboard. Mostly used as an animation layer.

```
public class VirtualKey : MonoBehaviour, IVirtualKeyboardRaycastTarget
```

Inheritance

[object](#)⤢ ← Object ← Component ← Behaviour ← MonoBehaviour ← VirtualKey

Implements
[IVirtualKeyboardRaycastTarget](#)

Extension Methods
[VirtualKeyboardUtilities.ApplyKeyPointersToChildren(VirtualKey)](#)

Fields

Character

The primary character of this key

```
public string Character
```

Field Value

[string](#)⤢

DisplaySecondaryKey

Whether not to display the primary and secondary key at the same time

```
public bool DisplaySecondaryKey
```

Field Value

[bool↗](#)

SecondaryCharacter

The secondary character for this key

```
public string SecondaryCharacter
```

Field Value

[string↗](#)

Methods

GetGridCoordinates()

Returns the grid coordinates relative to the keyboard layout

```
public Vector2 GetGridCoordinates()
```

Returns

Vector2

GetKey()

Returns the virtualkey object. Part of IVirtualKeyboardRaycastTarget implementation

```
public VirtualKey GetKey()
```

Returns

[VirtualKey](#)

Returns this virtual key

GetRectTransformPosition()

Returns the local position of this keys rect transform

```
public Vector2 GetRectTransformPosition()
```

Returns

Vector2

Initialize(VirtualKeyboardLayout, int, int)

Initializes the key and assignes all references. This is where layout reference is assigned and coordinates

```
public void Initialize(VirtualKeyboardLayout layout, int x, int y)
```

Parameters

`layout` VirtualKeyboardLayout

   The layout this key belongs to

x int⧉

   The x coordinate of this key

y int⧉

   The y coordinate of this key

Press()

Performs the Press animation for this key

```
public void Press()
```

Select()

Perform the Select animation logic

```
public void Select()
```

ToggleSelected(bool)

Toggles this key on or off

```
public void ToggleSelected(bool toggledOn)
```

Parameters

`toggledOn` [bool](#)

  On or Off

UnSelect()

Perform the Unselect animation logic

```
public void UnSelect()
```

UpdateShownKey(bool, bool)

Animates the shown key based on the state of capsLock and shift

```
public void UpdateShownKey(bool capsLockPressed, bool shiftPressed)
```

Parameters

`capsLockPressed` [bool](#)

  Whether capslock is toggled

`shiftPressed` [bool](#)

  Whether shift is toggled

Class VirtualKeyPointer

Namespace: LastAssets.LastVirtualKeyboard

Pointer class, useful for pointing any raycastable transform to a virtual key

```
public class VirtualKeyPointer : MonoBehaviour, IVirtualKeyboardRaycastTarget
```

Inheritance

object ⬀ ← Object ← Component ← Behaviour ← MonoBehaviour ← VirtualKeyPointer

Implements

IVirtualKeyboardRaycastTarget

Properties

Key

Sets the virtual key this object points to

```
public VirtualKey Key { set; }
```

Property Value

VirtualKey

Class VirtualKeyboard

Namespace: [LastAssets](.)[.LastVirtualKeyboard](.)

Main Virtual keyboard class, required for functionality

```
[RequireComponent(typeof(VirtualKeyboardLayout))]
public class VirtualKeyboard : MonoBehaviour
```

Inheritance

[object](.) ← Object ← Component ← Behaviour ← MonoBehaviour ← VirtualKeyboard

Fields

Layout

```
public VirtualKeyboardLayout Layout
```

Field Value

[VirtualKeyboardLayout](.)

RepeatInputTime

```
public float RepeatInputTime
```

Field Value

[float](.)

Methods

CastForKeyWithPointerData(EventSystem, PointerEventData)

Returns a VirtualKey under the pointer data position. Returns null if no key was found.

```
public VirtualKey CastForKeyWithPointerData(EventSystem eventSystem,
PointerEventData pointerData)
```

Parameters

`eventSystem` EventSystem

The referenced eventsystem to use

`pointerData` PointerEventData

The pointer data, make sure to set the position!

Returns

VirtualKey

DeselectFocusedKey()

If any key is currently selected, perform the unselect method on it

```
public void DeselectFocusedKey()
```

Hide()

Sets the object inactive

```
public void Hide()
```

IsOpen()

Returns true or false depending on if the object is active or inactive

```
public bool IsOpen()
```

Returns

bool ↗

OnCursorUpdated(EventSystem, PointerEventData)

Perform a cursor update, will cause hovering effect when over selected keys

```
public void OnCursorUpdated(EventSystem eventSystem, PointerEventData pointerData)
```

Parameters

eventSystem EventSystem

    The event system to read from

pointerData PointerEventData

    The data representing the simulated cursor. Make sure to assign its position!

OnNavigation(Vector2, bool)

Perform Up/Down/Left/Right movement relative to currently selected key

```
public void OnNavigation(Vector2 direction, bool performImmediately)
```

Parameters

direction Vector2

    The Up/Down/Left/Right relative to the currently selected key

performImmediately bool ↗

    Whether not to perform the movement immediately

PlayHoveredSound()

Force play a snippet of the layouts Hovered sound

```
public void PlayHoveredSound()
```

PlaySelectedSound()

Force play a snippet of the layouts Selected sound

```
public void PlaySelectedSound()
```

PressKey(string)

Performs a press on the target string

```
public void PressKey(string toPress)
```

Parameters

toPress string ↗

   The string representing the key to press

ReadKey()

Reads the currently selected virtual key and returns the string the key represents

```
public string ReadKey()
```

Returns

string ↗

Show()

Sets the object active

```
public void Show()
```

ToggleCapsLock()

State Toggle for CapsLock

```
public void ToggleCapsLock()
```

ToggleShift()

State Toggle for Shift

```
public void ToggleShift()
```

Class VirtualKeyboardLayout

Namespace: LastAssets.LastVirtualKeyboard

Layout component that creates and adjusts the virtual keyboard. This is a required component

```
[RequireComponent(typeof(RectTransform))]
public class VirtualKeyboardLayout : MonoBehaviour
```

Inheritance

object⬏ ← Object ← Component ← Behaviour ← MonoBehaviour ← VirtualKeyboardLayout

Fields

Layout

Layout reference

```
public VirtualKeyboardLayout.KeyboardLayout Layout
```

Field Value

VirtualKeyboardLayout.KeyboardLayout

_keyInstances

```
public List<VirtualKey> _keyInstances
```

Field Value

List⬏ <VirtualKey >

Methods

GetRandomKey()

Returns a random key, somewhere on the keyboard

```
public VirtualKey GetRandomKey()
```

Returns

VirtualKey

Move(VirtualKey, Vector2)

Performs a move in direction relative to inputted key.

```
public VirtualKey Move(VirtualKey referenceKey, Vector2 direction)
```

Parameters

referenceKey VirtualKey

   The key to move from

direction Vector2

   The direction to move to

Returns

VirtualKey

ToggleKeysOfType(bool, string)

Searches for keys that match the character, if they are found they are set to the toggle state

```
public void ToggleKeysOfType(bool isToggled, string character)
```

Parameters

`isToggled` [bool⧉](#)

  Whether its toggled on or not

`character` [string⧉](#)

  The character we want to search for

UpdateKeysWithState(bool, bool)

Updates the keys given the state of caps lock and shift

```
public void UpdateKeysWithState(bool capsLockPressed, bool shiftPressed)
```

Parameters

`capsLockPressed` [bool⧉](#)

  Is the keyboards capslock pressed

`shiftPressed` [bool⧉](#)

  Is the keyboards shift pressed

UpdateLayout()

Recreates the keyboard layout. High impact on performance, so call this only when necessary!

```
public void UpdateLayout()
```

Struct VirtualKeyboardLayout.AdditiveKeyIcon

Namespace: [LastAssets](#).[LastVirtualKeyboard](#)

Additional information to be passed into the creation of a keyboarditem. Can be text or an image

```
[Serializable]
public struct VirtualKeyboardLayout.AdditiveKeyIcon
```

Fields

Icon

```
public Sprite Icon
```

Field Value

Sprite

KeyRoundedness

```
public float KeyRoundedness
```

Field Value

[float](#)↗

KeyTileMode

```
public Image.Type KeyTileMode
```

Field Value

Image.Type

LocalScale

```
public Vector3 LocalScale
```

Field Value

Vector3

PositionOffset

```
public Vector2 PositionOffset
```

Field Value

Vector2

UIAnchor

```
public VirtualKeyboardUtilities.UIAnchor UIAnchor
```

Field Value

[VirtualKeyboardUtilities](#).[UIAnchor](#)

UseTextColor

```
public bool UseTextColor
```

Field Value

[bool](#)⤢

Struct VirtualKeyboardLayout.KeyboardItem

Namespace: LastAssets.LastVirtualKeyboard

An item that will show up on the keyboard

```
[Serializable]
public struct VirtualKeyboardLayout.KeyboardItem
```

Fields

AdditiveIcon

```
public VirtualKeyboardLayout.AdditiveKeyIcon AdditiveIcon
```

Field Value

VirtualKeyboardLayout.AdditiveKeyIcon

CellWidthMultiplier

```
public float CellWidthMultiplier
```

Field Value

float↗

DisplaySecondaryKey

```
public bool DisplaySecondaryKey
```

Field Value

bool↗

Key

```
public string Key
```

Field Value

[string](#)↗

SecondaryKey

```
public string SecondaryKey
```

Field Value

[string](#)↗

UseAdditiveIcon

```
public bool UseAdditiveIcon
```

Field Value

[bool](#)↗

Class VirtualKeyboardLayout.KeyboardLayout

Namespace: LastAssets.LastVirtualKeyboard

Main components to the keyboard, customization can be done here

```
[Serializable]
public class VirtualKeyboardLayout.KeyboardLayout
```

Inheritance

object ← VirtualKeyboardLayout.KeyboardLayout

Fields

AllRows

```
public List<VirtualKeyboardLayout.KeyboardRow> AllRows
```

Field Value

List <VirtualKeyboardLayout.KeyboardRow>

AudioSource

```
public AudioSource AudioSource
```

Field Value

AudioSource

BackgroundColor

```
public Color BackgroundColor
```

Field Value

Color

ExternalPadding

```
public float ExternalPadding
```

Field Value

[float](#)⧉

Font

```
public TMP_FontAsset Font
```

Field Value

TMP_FontAsset

HighlightTransitionTime

```
public float HighlightTransitionTime
```

Field Value

[float](#)⧉

HighlightedColor

```
public Color HighlightedColor
```

Field Value

Color

HoveredKeyClip

```
public AudioClip HoveredKeyClip
```

Field Value

AudioClip

KeyColor

```
public Color KeyColor
```

Field Value

Color

KeyFocusTransitionTime

```
public float KeyFocusTransitionTime
```

Field Value

[float↗](#)

KeyImage

```
public Sprite KeyImage
```

Field Value

Sprite

KeyPadding

```
public float KeyPadding
```

Field Value

[float⌝](#)

KeyRoundedness

```
public float KeyRoundedness
```

Field Value

[float⌝](#)

KeyTileMode

```
public Image.Type KeyTileMode
```

Field Value

Image.Type

KeyboardImage

```
public Sprite KeyboardImage
```

Field Value

Sprite

NavigationWrapAroundHorizontal

```
public bool NavigationWrapAroundHorizontal
```

Field Value

[bool⌤](#)

PressAnimationTime

```
public float PressAnimationTime
```

Field Value

[float⌤](#)

PressColor

```
public Color PressColor
```

Field Value

Color

SecondaryTextColor

```
public Color SecondaryTextColor
```

Field Value

Color

SelectedKeyClip

```
public AudioClip SelectedKeyClip
```

Field Value

AudioClip

StackedTextScale

```
public float StackedTextScale
```

Field Value

[float](#)

TextColor

```
public Color TextColor
```

Field Value

Color

TextScale

```
public float TextScale
```

Field Value

[float](#)

ToggledButtonColor

```
public Color ToggledButtonColor
```

Field Value

Color

Class VirtualKeyboardLayout.KeyboardRow

Namespace: LastAssets.LastVirtualKeyboard

Keyboard row, holds a list of KeyboardItems that get aligned horizontally

```
[Serializable]
public class VirtualKeyboardLayout.KeyboardRow
```

Inheritance

object ← VirtualKeyboardLayout.KeyboardRow

Fields

Keys

```
public List<VirtualKeyboardLayout.KeyboardItem> Keys
```

Field Value

List < VirtualKeyboardLayout.KeyboardItem >

Struct VirtualKeyboardLayout.VirtualHotkey

Namespace: LastAssets.LastVirtualKeyboard

Binds an inputsystem action to a key result

```
[Serializable]
public struct VirtualKeyboardLayout.VirtualHotkey
```

Fields

ActionToBind

```
public InputActionReference ActionToBind
```

Field Value

InputActionReference

Key

```
public string Key
```

Field Value

string↗

Class VirtualKeyboardUtilities

Namespace: [LastAssets](.)[.LastVirtualKeyboard](.)

VirtualKeyboard Utility class

```
public static class VirtualKeyboardUtilities
```

Inheritance

[object](.)⬀ ← VirtualKeyboardUtilities

Methods

ApplyKeyPointersToChildren(VirtualKey)

Applies the VirtualKey component to every child of the target

```
public static void ApplyKeyPointersToChildren(this VirtualKey target)
```

Parameters

target [VirtualKey](.)

TranslateUIAnchorToPivot(UIAnchor)

Converts the enum UIAnchor to a Vector2 pivot point used in the unity UI system

```
public static Vector2 TranslateUIAnchorToPivot(VirtualKeyboardUtilities.UIAnchor anchor)
```

Parameters

anchor [VirtualKeyboardUtilities](.)[.UIAnchor](.)

Returns

Vector2

Enum VirtualKeyboardUtilities.UIAnchor

Namespace: [LastAssets](#).[LastVirtualKeyboard](#)

Repesents a Vector2 pivot point used in UnityEngines UI RectTransform

```
public enum VirtualKeyboardUtilities.UIAnchor
```

Fields

LowerCenter = 7

LowerLeft = 6

LowerRight = 8

MiddleCenter = 4

MiddleLeft = 3

MiddleRight = 5

UpperCenter = 1

UpperLeft = 0

UpperRight = 2

Namespace LastAssets.LastVirtualKeyboard. Examples

Classes

[EXAMPLE_InputBinding](#)

Example class for connecting Unity's new input system to the keyboard

Class EXAMPLE_InputBinding

Namespace: [LastAssets](#).[LastVirtualKeyboard](#).[Examples](#)

Example class for connecting Unity's new input system to the keyboard

```
public class EXAMPLE_InputBinding : MonoBehaviour
```

Inheritance

[object](#)⧉ ← Object ← Component ← Behaviour ← MonoBehaviour ← EXAMPLE_InputBinding

Fields

EventSystemRef

Reference to a unity eventsystem reference

```
public EventSystem EventSystemRef
```

Field Value

EventSystem

InputModule

Input system generated input class

```
public InputSystem_VirtualKeyboard InputModule
```

Field Value

[InputSystem_VirtualKeyboard](#)

InputSystemUIRef

Reference to unity's InputSystemUIRef

```
public InputSystemUIInputModule InputSystemUIRef
```

Field Value

InputSystemUIInputModule

VirtualKeyboardRef

Reference to the virtual keyboard

```
public VirtualKeyboard VirtualKeyboardRef
```

Field Value

VirtualKeyboard