Clicky Controller

Sean Edwards

Version 1.0.0

Fri May 1 2020

# Hierarchical Index

## Class Hierarchy

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

ClickyController.Controller 6

ClickyController.Keyboard 9

ClickyController.Mouse 13

ClickyController.Mouse.POINT 19

# Class Index

## Class List

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

**ClickyController.Controller (Contains the methods and structs that are needed by the Mouse and Keyboard classes to send input to Windows. This class also contains any miscellaneous features that don't necessarily fit under the function of a 'mouse' or 'keyboard )** 6

**ClickyController.Keyboard (Contains the functionality that would be associated with using a keyboard )** 9

**ClickyController.Mouse (Contains the functionality that would typically be associated with the mouse e.g mouse clicks, scrolling and moving the mouse )** 13

**ClickyController.Mouse.POINT (Holds the information about the mouse X and Y coordinates )** 19

# Namespace Documentation

## ClickyController Namespace Reference

### Classes

class **Controller**

*Contains the methods and structs that are needed by the* ***Mouse*** *and* ***Keyboard*** *classes to send input to Windows. This class also contains any miscellaneous features that don't necessarily fit under the function of a 'mouse' or 'keyboard.*

class **Keyboard**

*Contains the functionality that would be associated with using a keyboard*

class **Mouse**

*Contains the functionality that would typically be associated with the mouse e.g mouse clicks, scrolling and moving the mouse*

# Class Documentation

## ClickyController.Controller Class Reference

A screenshot of a cell phone

Description automatically generated

Contains the methods and structs that are needed by the **Mouse** and **Keyboard** classes to send input to Windows. This class also contains any miscellaneous features that don't necessarily fit under the function of a 'mouse' or 'keyboard.

Inherited by **ClickyController.Keyboard**, and **ClickyController.Mouse**.

### Classes

struct **INPUT**

*Contains the information needed by 'SendInput' function to create a simulted mouse/keyboard event More information can be found here https://docs.microsoft.com/en-us/windows/win32/api/winuser/ns-winuser-input*

struct **InputUnion**

*A workaround for the lack of a native 'Union' function in C#*

struct **KEYBDINPUT**

*Contains information about a simulated keyboard event More information can be found here https://docs.microsoft.com/en-gb/windows/win32/api/winuser/ns-winuser-keybdinput*

struct **MOUSEINPUT**

*Contains information about a simulated mouse event More information can be found here https://docs.microsoft.com/en-us/windows/win32/api/winuser/ns-winuser-mouseinput*

### Static Public Member Functions

static void **Wait** (int seconds)

*Waits the inputted amount of time in seconds*

### Detailed Description

Contains the methods and structs that are needed by the **Mouse** and **Keyboard** classes to send input to Windows. This class also contains any miscellaneous features that don't necessarily fit under the function of a 'mouse' or 'keyboard.

### Member Function Documentation

#### static void ClickyController.Controller.Wait (int *seconds*)[static]

Waits the inputted amount of time in seconds

##### Parameters

|  |  |
| --- | --- |
| *seconds* | Number of seconds to pause the program for |

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

W:/Year 3/Project/ClickyController/ClickyController/Controller.cs

## ClickyController.Keyboard Class Reference

A screenshot of a cell phone

Description automatically generated

Contains the functionality that would be associated with using a keyboard

Inherits **ClickyController.Controller**.

### Static Public Member Functions

static void **KeyPress** (string character)

*Performs a simple key press*

static void **EnterText** (string textEntry)

*Will 'type' the user specified string*

static void **KeyDown** (string character)

*Holds a key down using it's virtual key*

static void **KeyUp** (string character)

*Releases a key*

static void **KeyboardShortcut** (string character1, string character2, string character3="")

*Performs a keyboard shortcut by pressing all of the keys at the same time*

static void **KeyboardShortcutScanCode** (string character1, string character2, string character3="")

*Performs a keyboard shortcut by pressing all of the keys at the same time, using the key's scan codes*

static void **KeyPressScanCode** (string character)

*Performs a key press using the key's scan code. ScanCodes are the codes sent directly by your keyboard hardware and can be useful in apps/games that take their input directly from the keyboard directly.*

static void **KeyDownScanCode** (string character)

*Holds a key down using its scan code*

static void **KeyUpScanCode** (string character)

*Releases a key using its scan code*

static bool **VirtualCodeKeyExists** (string character)

*Checks whether a key or character has a Virtual Code stored in the Virtual Key dictionary*

static bool **ScanCodeKeyExists** (string character)

*Checks whether a key or character has a Scan Code stored in the Scan Key dictionary*

### Static Private Attributes

static readonly Dictionary< string, ushort > **KeyToVirtualKeyDictionary** = JsonConvert.DeserializeObject<Dictionary<string, ushort>>(Properties.Resources.VirtualKeyCodes)

static readonly Dictionary< string, string > **KeyToVirtualKeyShiftDictionary** = JsonConvert.DeserializeObject<Dictionary<string, string>>(Properties.Resources.VirtualKeyCodesShift)

static readonly Dictionary< string, ushort > **KeyToScanCodeDictionary** = JsonConvert.DeserializeObject<Dictionary<string, ushort>>(Properties.Resources.ScanCodes)

### Detailed Description

Contains the functionality that would be associated with using a keyboard

### Member Function Documentation

#### static void ClickyController.Keyboard.EnterText (string *textEntry*)[static]

Will 'type' the user specified string

##### Parameters

|  |  |
| --- | --- |
| *textEntry* | The text that the user wishes to type out on the screen |

#### static void ClickyController.Keyboard.KeyboardShortcut (string *character1*, string *character2*, string *character3* = "")[static]

Performs a keyboard shortcut by pressing all of the keys at the same time

##### Parameters

|  |  |
| --- | --- |
| *character1* | Name of the first key to press down |
| *character2* | Name of the second key to press down |
| *character3* | Optional: Name of the third key to press down |

#### static void ClickyController.Keyboard.KeyboardShortcutScanCode (string *character1*, string *character2*, string *character3* = "")[static]

Performs a keyboard shortcut by pressing all of the keys at the same time, using the key's scan codes

##### Parameters

|  |  |
| --- | --- |
| *character1* | Name of the first key to press down |
| *character2* | Name of the second key to press down |
| *character3* | Optional: Name of the third key to press down |

#### static void ClickyController.Keyboard.KeyDown (string *character*)[static]

Holds a key down using it's virtual key

##### Parameters

|  |  |
| --- | --- |
| *character* | The key to hold down |

#### static void ClickyController.Keyboard.KeyDownScanCode (string *character*)[static]

Holds a key down using its scan code

##### Parameters

|  |  |
| --- | --- |
| *character* | Name of the key or character the user wishses to hold down |

#### static void ClickyController.Keyboard.KeyPress (string *character*)[static]

Performs a simple key press

##### Parameters

|  |  |
| --- | --- |
| *character* | The name of the key or the letter the user wishes to press |

#### static void ClickyController.Keyboard.KeyPressScanCode (string *character*)[static]

Performs a key press using the key's scan code. ScanCodes are the codes sent directly by your keyboard hardware and can be useful in apps/games that take their input directly from the keyboard directly.

##### Parameters

|  |  |
| --- | --- |
| *character* | Name of the key or character the user wishses to press |

#### static void ClickyController.Keyboard.KeyUp (string *character*)[static]

Releases a key

##### Parameters

|  |  |
| --- | --- |
| *character* | The key to release |

#### static void ClickyController.Keyboard.KeyUpScanCode (string *character*)[static]

Releases a key using its scan code

##### Parameters

|  |  |
| --- | --- |
| *character* | Name of the key or character the user wishses to release |

#### static bool ClickyController.Keyboard.ScanCodeKeyExists (string *character*)[static]

Checks whether a key or character has a Scan Code stored in the Scan Key dictionary

##### Parameters

|  |  |
| --- | --- |
| *character* | Name of the key or character the user wishses to check |

##### Returns

#### static bool ClickyController.Keyboard.VirtualCodeKeyExists (string *character*)[static]

Checks whether a key or character has a Virtual Code stored in the Virtual Key dictionary

##### Parameters

|  |  |
| --- | --- |
| *character* | Name of the key or character the user wishses to check |

##### Returns

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

W:/Year 3/Project/ClickyController/ClickyController/Keyboard.cs

## ClickyController.Mouse Class Reference

A screenshot of a cell phone

Description automatically generated

Contains the functionality that would typically be associated with the mouse e.g mouse clicks, scrolling and moving the mouse

Inherits **ClickyController.Controller**.

### Classes

struct **POINT**

*Holds the information about the mouse X and Y coordinates*

### Static Public Member Functions

static void **MoveMouse** (int xPosition, int yPosition, bool relativeToMousePosition=false)

*Moves mouse to a certain position on the screen*

static void **LeftClick** ()

*Performs a left click*

static void **LeftDown** ()

*Holds to the left button down*

static void **LeftUp** ()

*Lifts the left button up*

static void **RightClick** ()

*Performs a right click*

static void **RightDown** ()

*Holds the right button down*

static void **RightUp** ()

*Lifts the right button up*

static void **MiddleClick** ()

*Performs a click with the middle button (the scroll wheel)*

static void **MiddleDown** ()

*Holds the middle mouse button down*

static void **MiddleUp** ()

*Lifts the middle mouse button up*

static void **ScrollDown** (uint clicks)

*Scrolls the mouse wheel down*

static void **ScrollUp** (uint clicks)

*Scrolls the mouse wheel up*

### Properties

static **POINT** **MousePosition** [get, set]

*Holds information about the current position of the mouse*

static int **XCoordinate** [get, private set]

*Returns the current X coordinate of the mouse*

static int **YCoordinate** [get, private set]

*Returns the current Y coordinate of the mouse*

### Private Member Functions

static bool **GetCursorPos** (out **POINT** mousePosition)

*Windows API that returns the position of the cursor with its X and Y coordinates*

static bool **SetCursorPos** (int x, int y)

*Windows API that moves the mouse to the given X/Y coordinates*

### Static Private Member Functions

static void **MouseAction** (uint mouseActionCode, uint mouseData=0)

*Performs a mouse action (that is, either Down or Release). This allows a user to perform an action like dragging the mouse or long button presses.*

### Static Private Attributes

static **POINT** **\_mousePosition**

### Detailed Description

Contains the functionality that would typically be associated with the mouse e.g mouse clicks, scrolling and moving the mouse

### Member Function Documentation

#### static bool ClickyController.Mouse.GetCursorPos (out POINT *mousePosition*)[private]

Windows API that returns the position of the cursor with its X and Y coordinates

##### Parameters

|  |  |
| --- | --- |
| *mousePosition* | **POINT** sruct containing the X and Y coordinates of the mouse |

##### Returns

#### static void ClickyController.Mouse.LeftClick ()[static]

Performs a left click

#### static void ClickyController.Mouse.LeftDown ()[static]

Holds to the left button down

#### static void ClickyController.Mouse.LeftUp ()[static]

Lifts the left button up

#### static void ClickyController.Mouse.MiddleClick ()[static]

Performs a click with the middle button (the scroll wheel)

#### static void ClickyController.Mouse.MiddleDown ()[static]

Holds the middle mouse button down

#### static void ClickyController.Mouse.MiddleUp ()[static]

Lifts the middle mouse button up

#### static void ClickyController.Mouse.MouseAction (uint *mouseActionCode*, uint *mouseData* = 0)[static], [private]

Performs a mouse action (that is, either Down or Release). This allows a user to perform an action like dragging the mouse or long button presses.

##### Parameters

|  |  |
| --- | --- |
| *mouseActionCode* | HEX code associated with the mouse button and the state for it to be in (Up or Down) |
| *mouseData* | Used by the Scroll functions and represents how many clicks of the scroll wheel they wish to do. A negative value scrolls down and a positive value scrolls up. |

#### static void ClickyController.Mouse.MoveMouse (int *xPosition*, int *yPosition*, bool *relativeToMousePosition* = false)[static]

Moves mouse to a certain position on the screen

##### Parameters

|  |  |
| --- | --- |
| *xPosition* | The X coordinate to move the mouse to. |
| *yPosition* | The Y coordinate to move the mosue to. |
| *relativeToMousePosition* | Optional paramter: If enabled, will move the mouse relative to its current position on the screen |

#### static void ClickyController.Mouse.RightClick ()[static]

Performs a right click

#### static void ClickyController.Mouse.RightDown ()[static]

Holds the right button down

#### static void ClickyController.Mouse.RightUp ()[static]

Lifts the right button up

#### static void ClickyController.Mouse.ScrollDown (uint *clicks*)[static]

Scrolls the mouse wheel down

##### Parameters

|  |  |
| --- | --- |
| *clicks* | The number of mouse wheel 'clicks' to scroll down by |

#### static void ClickyController.Mouse.ScrollUp (uint *clicks*)[static]

Scrolls the mouse wheel up

##### Parameters

|  |  |
| --- | --- |
| *clicks* | The number of mouse wheel 'clicks' to scroll up by |

#### static bool ClickyController.Mouse.SetCursorPos (int *x*, int *y*)[private]

Windows API that moves the mouse to the given X/Y coordinates

##### Parameters

|  |  |
| --- | --- |
| *x* | The X coordinate to move the mouse to |
| *y* | The Y coordinate to move the mouse to |

##### Returns

### Property Documentation

#### POINT ClickyController.Mouse.MousePosition[static], [get], [set], [private]

Holds information about the current position of the mouse

#### int ClickyController.Mouse.XCoordinate[static], [get], [private set]

Returns the current X coordinate of the mouse

#### int ClickyController.Mouse.YCoordinate[static], [get], [private set]

Returns the current Y coordinate of the mouse

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

W:/Year 3/Project/ClickyController/ClickyController/Mouse.cs

## ClickyController.Mouse.POINT Struct Reference

Holds the information about the mouse X and Y coordinates

### Detailed Description

Holds the information about the mouse X and Y coordinates

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

W:/Year 3/Project/ClickyController/ClickyController/Mouse.cs