Tuxis Input System

Documentation

2012

CLASS INPUT

Input class is responsible for processing the input in the window or in the entire workspace.

Main Header: « **input.h** »

Static library: « **InputModule.lib** »

Dynamic library: «**InputModule.dll** »

Methods:

Input(HWND pWindowHandler **);**
Parameters
Constructor. Is used to initialize the DirectInput device.
Description
pWindowHandler – identifier of window in Windows OS. If you pass a null value then the class will be operate within the entire working area of the screen.

void **Update();**
Parameters
None.
Description
Updating states of input devices.

bool **KeyDown(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace)

Description

This method is used to detect if a key is being held down.

This method returns a **false** if the key is not held down, a **true** if the key is held down.

bool **KeyHit(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace)

Description

This method is used to detect a hit of specified key, from last call this method.

This method returns a **false** if the key is has not been hit down, a **true** if the key has been hit down.

bool **KeyUp(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace)

Description

This method is used to detect a up(release) of specified key, from last call this method.

This method returns a **false** if the key is has not been released, a **true** if the key has been released.

bool **MouseDown(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace (MOUSE_ prefix))

Description

This method is used to detect if a mouse key is being held down.

This command returns a **false** if the key is not held down, a **true** if the key is held down.

bool **MouseHit(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace (MOUSE_ prefix))

Description

This method is used to detect a hit of specified mouse key, from last call this method.

This method returns a **false** if the key is has not been hit down, a **true** if the key has been hit down.

bool **MouseUp(** BYTE key **);**

Parameters

key – Keycode (See Key's namespace (MOUSE_ prefix))

Description

This method is used to detect a up(release) of specified mouse key, from last call this method.

This method returns a **false** if the key is has not been released, a **true** if the key has been released.

int **MouseSpeedX();**
int **MouseSpeedY();**
int **MouseSpeedZ();**

Parameters

None.

Description

MouseSpeedX & **MouseSpeedY** methods find the difference between where the mouse WAS to where it is NOW, from last call **Update** method.
MouseZSpeed returns value if the mousewheel on a suitable mouse is being rolled, 0 if it is not being moved.

void **SetMousePosition**(int x, int y);
Parameters
x, y – cursor coordinates.
Description
Use this method to move the mouse to a designated location.

void **GetMousePosition**(int& x, int& y);
Parameters
x, y – cursor coordinates.
Description
This method writes to the x & y variables a location of the mouse on the window (screen)