Tuxis Input System

Documentation

2012

LIB INFO:

Main Header: « **InputModule.h** »

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

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

CLASS: INPUT MANAGER

InputManager (HWND pWindowHandler);

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.

IMouse GetMouse();

Description

Getting the mouse device interface. IMouse object can’t be created manually.

IKeyboard GetKeyboard()

Description

Getting the keyboard device interface. IKeyboard object can’t be created manually.

void UpdateAllStates();

Description

Update states of all input interfaces. (You can do this manually by calling .Update() for every input object)

CLASS: IKEYBOARD

bool ButtonDown(BYTE mouse_ keycode);

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 ButtonHit(BYTE mouse_ keycode);

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 ButtonUp(BYTE mouse_keycode);

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.

void Update();

Description

Update keyboard device state.

CLASS: IMOUSE

bool ButtonDown(BYTE mouse_ keycode);

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 ButtonHit(BYTE mouse_ keycode);

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 ButtonUp(BYTE mouse_keycode);

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.

void Update();

Description

Update keyboard device state.

int SpeedX();
int SpeedY();
int SpeedZ();

Description

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

void **SetCursorPosition**(int x, int y);

Parameters

x, y – cursor coordinates.

Description

Use this method to move the mouse to a designated location.

void **GetCursorPosition**(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)