- C

Type text to search here...

> D > 1: A

XInput Tutorial Part 1: Adding gamepad support to your Windows game

A 30, 2013
12

C++, A D 11.
D D
, - .
C 360 :
360 , B C
.

(, 2D
1.11),
, A .

, :

NOTE: If you can't be bothered with the low-level details and just want to shoehorn gamepad support into a game really quickly, my Simple2D library (version 1.11 and above) includes gamepad support – see the Tetris gamepad support article for a quick example on how to use the library to add gamepad support in just a few minutes!

Setting up your build environment

; 1.3
7, 1.4 8 9.1.0
, .
9.1.0 . , *#include <xinput.h>*
, *Xinput9_1_0.lib* (
1.3/1.4, *XInput.lib*).

How to check whether a controller is connected

0-3 (
360 ,
, 0 - ,
).

, ,
. , ,
.

XInputGetState .
, *XINPUT_STATE* . / .

```
XINPUT_STATE state;
ZeroMemory(&state, si    (XINPUT_STATE));

if (XInputGetState(0, &state) == ERROR_SUCCESS)
{
// controller is connected on port 0
}
```

0.
, :

```
int controllerId = -1;

for (DWORD i = 0; i <          _COUNT && controllerId == -1; i++)
{
XINPUT_STATE state;
ZeroMemory(&state, si    (XINPUT_STATE));

if (XInputGetState(i, &state) == ERROR_SUCCESS)
controllerId = i;
}
```

for , *controllerId*
, -1 .

How to test whether a particular button is pressed down

XINPUT_STATE *XINPUT_GAMEPAD* ,
wButtons 1
, 0 . *wButtons*
:

F

```
XINPUT_GAMEPAD_A
XINPUT_GAMEPAD_B
XINPUT_GAMEPAD_X
XINPUT_GAMEPAD_Y
```

D

```
XINPUT_GAMEPAD_DPAD_LEFT
XINPUT_GAMEPAD_DPAD_RIGHT
XINPUT_GAMEPAD_DPAD_UP
XINPUT_GAMEPAD_DPAD_DOWN
```

```
XINPUT_GAMEPAD_LEFT_SHOULDER
XINPUT_GAMEPAD_RIGHT_SHOULDER
```

A ( )

```
XINPUT_GAMEPAD_LEFT_THUMB
XINPUT_GAMEPAD_RIGHT_THUMB
```

C

```
XINPUT_GAMEPAD_BACK
XINPUT_GAMEPAD_START
```

, A D
wButtons . F , A,
:

```
bool A_button_pressed = ((state.Gamepad.wButtons & XINPUT_GAMEPAD_A
```

How to test how much each trigger is depressed

360
(
).
8- (0-255 0
255).
:

```
float leftTrigger = (float) state.Gamepad.bLeftTrigger / 255;
float rightTrigger = (float) state.Gamepad.bRightTrigger / 255;
```

0-1
.
state.Gamepad.bLeftTrigger *state.Gamepad.bRightTrigger* .

How to test the position of each analog thumb stick

, ( ) (0,0).
, ,
.
16- (-32768 +32767 -32768
, 0 32767 ,
). :

```
float normLX = fmaxf(-1, (float) state.Gamepad.sThumbLX / 32767);
float normLY = fmaxf(-1, (float) state.Gamepad.sThumbLY / 32767);
```

(versions of Visual Studio prior to Visual Studio 2013 use a <math.h> header which isn't C99-compliant. If fmaxf() doesn't work in your compiler, try using max() instead)

state.Gamepad.sThumbLX,
state.Gamepad.sThumbLY ( ) *state.Gamepad.sThumbRX,*
state.Gamepad.sThumbRY ( ).
.

: -1 1.
32768, 1
32767. , 32767 ,
( ) -1.
-1 -1
+1.

, :
, , (
, -400 400
). ,
. ,
deadzone .

How to calibrate deadzones for the analog sticks

A *deadzone* (
,)
. -
,
(
).
- ,
.

-1 +1 ,
:

```
float deadzoneX = 0.05f;
float deadzoneY = 0.02f;

float leftStickX = (abs(normLX) < deadzoneX ? 0 : normLX);
float leftStickY = (abs(normLY) < deadzoneY ? 0 : normLY);
```

- (5% - 2% - ,
360
).
, ,
.

, . 80%.
80% , 0
. B 80%-100%, 80%-100%,
-
.
( )
0-1. F , 80%, 80% 0, 85%
0.25, 90% 0.5 .

, :

```
leftStickX = (abs(normLX) < deadzoneX ? 0 : (abs(normLX) - deadzone
leftStickY = (abs(normLY) < deadzoneY ? 0 : (abs(normLY) - deadzone

if (deadzoneX > 0) leftStickX /= 1 - deadzoneX;
if (deadzoneY > 0) leftStickY /= 1 - deadzoneY;
```

E , ,
( ) .
80%, 0-0.2 (0%-20%).
( ) , *x / abs(x)* 1
-1 . F , *if* 0
1. 80% , 0-0.2 1
0.8 = 0.2, 5. C , 0.2 (

80%) * 5 = 1,

.

, .

Rolling it all together into a Gamepad class

:

+ expand source

A , *Gamepad*. ( ), *Refresh()*.

, *Refresh()*. .

A *Refresh()*, *GetPort(), GetState(), IsPressed(WORD button)* *leftStickX, leftStickY, rightStickX, rightStickY, leftTrigger rightTrigger* / . *Gamepad*

, ,

.

- :

```
#include <iostream>
#include <Windows.h>
#include <Xinput.h>

using std::cout;
using std::endl;

/* Insert the Gamepad class code here */

int main()
{
Gamepad gamepad;

bool wasConnected = true;

while (true)
{
Sleep(100);

if (!gamepad.Refresh())
{
if (wasConnected)
{
wasConnected = false;

cout << "Please connect an Xbox 360 controller." << endl;
}
}
else
{
if (!wasConnected)
{
wasConnected = true;

cout << "Controller connected on port " << gamepad.GetPort() << enc
}
```

```
cout << "Left thumb stick: (" << gamepad.leftStickX << ", " << game

cout << "Left analog trigger: " << gamepad.leftTrigger << "   Right

if (gamepad.IsPressed(XINPUT_GAMEPAD_A)) cout << "(A) button presse
}
}
}
```

Wait, is that it?!

, . A ,
. *GetAsyncKeyState()*
, .
Gamepad::IsPressed()
. ,

.

(*WM_KEYDOWN* *WM_KEYUP*) .
- ,
, .
/ .
,
, ,
.

, 2 - .

, !

I'm a software developer with very limited work capacity due to having the debilitating illness M.E. – please read my article Dying with M.E. as a Software Developer and donate to the crowdfund to help me with my bucket list if you found this article useful. Thank you so much!

Useful Links

C A @ D A
,