Play with any USB controller on Linux using xboxdrv to emulate a XBOX controller

EDIT: This tutorial works for games you play from Steam Client and for games you play outside of the Steam Client. If you're willing to play only Steam Games, try this Steam Big Picture tutorial first. If it doesn't work, then xboxdrv is guaranteed to put your controller to work.

Some games are meant to be played using a XBOX controller by default. If you have one, excelent, but if you don't, some controllers won't work 100% out of the box. For example, there are games where a PS2 controller + USB adapter will missplace some buttons: A is where Y should be, X is where A should be etc. You can correct this by remapping all your buttons/axis to the right place. On Windows you would use a program called **x360ce** and on Linux you'll use **xboxdrv**. I'll explaing how to set **xboxdrv** for your USB controller on Linux.

1. Installing xboxdrv

If your distro already has **xboxdrv** packed, install it from the repositories. If it doesn't, download the source code from their website (http://pingus.seul.org/~grumbel/xboxdrv/) and compile it. Since this step is distro specific, I won't cover it. If you're not sure how to do it, ask for help in your distro's forum.

2. Avoiding xpad conflict

Check if your distro loads **xpad** module to handle XBOX controller events:

```
# lsmod | grep xpad
```

If it returns a blank line, you're fine, go straight to the next section. If it returns something else, you must unload xpad module before loading xboxdrv by executing this command:

```
# rmmod xpad
```

3. Finding the proper input event

List all your available input events:

```
# ls /dev/input/ | grep event*
```

It will probably range from event0 to event20. You'll have to test each one of them until you find out which one is the event associated with your USB controller. To do so, enter the following command and press your controller buttons (press CTRL+C to exit after checking):

```
# evtest /dev/input/event11
```

If you got the wrong event, nothing will happen when you press buttons. When you get the right event, you'll notice because as soon as you press a button there will be a corresponding terminal output. Keep testing your available events until you find out which one is correct. In my case, */dev/input/event11* is associated with my PS2 controller.

4. Mapping your USB controller

If you have a PS3 controller, you can skip this section and go stragth to section 5. If you have any other controller, you must map it. To do so, while using evtest with the event associated to your controller (for example, # evtest /dev/input/event11), each time you press a button you'll receive a terminal output like this:

```
Event: time 1380985017.964843, type 4 (EV_MSC), code 4 (MSC_SCAN), value 90003
Event: time 1380985017.964843, type 1 (EV_KEY), code 290 (BTN_THUMB2), value 1
```

In this case, I pressed the button corresponding to where the A button is on the XBOX controller, and this button is mapped as**BTN_THUMB2**. Then I pressed the button corresponding to where the B button is on the XBOX controller, and the output was this:

```
Event: time 1380985018.460841, type 4 (EV_MSC), code 4 (MSC_SCAN), value 90002
Event: time 1380985018.460841, type 1 (EV_KEY), code 289 (BTN_THUMB), value 1
```

This output tells me that this button is mapped with the name**BTN_THUMB**. Take note of all these names. Do this for all your buttons and all your axis. You must take note of their names to be able to map them properly later. In the end, you'll have a list with all names and the corresponding XBOX buttons. You must map these buttons to valid XBOX buttons symbols:

Buttons: A, B, X, Y, RB (frontal upper right), RT (posterior upper right), LB (frontal upper left), LT (posterior upper left), START, BACK

Directionals: DPAD_X (horizontal D-pad), DPAD_Y (vertical D-pad), X1 (left analog horizontal), Y1 (left analog vertical), X2 (right analog horizontal), Y2 (right analog vertical)

Analog clicks: TL (left analog click), TR (right analog click)

I made an image to illustrate these valid XBOX buttons symbols [lh4.googleusercontent.com] you'll use to map your controller events. For a PS2 + USB adapter, this is how the final mapping list will look:

```
BTN_THUMB2=a
BTN_THUMB=b
BTN_BASE3=back
BTN_BASE4=start
BTN_BASE=lb
BTN_BASE2=rb
BTN_TOP2=lt
```

```
BTN_PINKIE=rt
BTN_BASE5=tl
BTN_BASE6=tr
ABS_X=x1
BTN_TOP=x
BTN_TRIGGER=y
ABS_Y=y1
ABS_RZ=x2
ABS_Z=y2
ABS_HAT0X=dpad_x
ABS_HAT0Y=dpad_y
-Y1=Y1
-Y2=Y2
```

Note that if you have a PS2 controller too, you won't have to map all your buttons again, since I already did this (just copy this list for further use). To see what each XBOX button is named after, you can use the built in xboxsrv help to see the valid names:

```
$ xboxsrv --help-button
$ xboxsrv --help-axis
$ xboxsrv --help-abs
```

5. Initializing xboxdrv

If you have a PS3 controller, you don't have to map your controller nor nothing. Just initialize xboxdrv like this and everything will be working:

```
# xboxdrv --silent --detach-kernel-driver
```

If you have any other controller, now that you have all your buttons and axis mapped, you must initialize xboxdrv properly. To do so, you'll have to initialize it like this:

```
# xboxdrv --evdev [EVENT] --evdev-absmap [ABS MAP] --axismap [AXIS MAP] --evdev-keymap [BUTTONS MAP] --mimic-xpad --silent &
```

[EVENT] is the event associated with your controller (section 3 of this post) and [ABS MAP], [AXIS MAP] and [BUTTONS MAP] are your controller mapping (section 4 of this post). In my case, my PS2 controller + USB adapter is associated with /dev/input/event11 and has the above mapping, so I initialize xboxdrv like this:

```
# xboxdrv --evdev /dev/input/event11 --evdev-absmap ABS_X=x1,ABS_Y=y1,ABS_RZ=x2,ABS_Z=y2,ABS_HAT0X=dpad_x,ABS_HAT0Y=dpad_y --axismap -Y1=Y1,-Y2=Y2 --evdev-keymap BTN_TOP=x,BTN_TRIGGER=y,BTN_THUMB2=a,BTN_THUMB=b,BTN_BASE3=back,BTN_BASE4=start,BTN_BASE=lb,BTN_BASE2=rb,BTN_TOP2=lt,BTN_PINKIE=rt,BTN_BASE5=tl,BTN_BASE6=tr --mimic-xpad --silent &
```

Note that if you have a PS2 controller too, you will initialize xboxdrv exactly the same way I do, except for the event, which might be another one.

6. Initializing xboxdrv during system startup

Everytime you restart your computer, you must unload xpad module (section 2) if it's loaded in your distro and properly initialize xboxdrv (section 5). To do these things automatically, you can put them on*/etc/rc.local* or whatever your distro calls it. It's distro specific, so go to your distro's forum and ask them where is your */etc/rc.local* if you can't find it. In my case, I use OpenSUSE 13.1 and this file is located at */etc/rc.d/boot.local*

7. Final words

I hope this helps you. With these steps I was able to set my USB controller to be remaped and emulated as a XBOX controller. I would appreciate if some moderator fixed this topic, so other users could easily find it.