Analog Joystick / Gamepad Control Sticks

Orx supports joysticks and gamepads. The axis (for an analog stick) usually has a value between 0 and 32768 which indicates how far along the axis the stick has been pushed. Orx scales these values to down to the range `0.0` to `1.0`.

This gives you smooth multi-direction control in both the X and Y directions.

Axis

Orx provides many Axis properties for your `Input` config.

Here's the list of the available joystick axes for the input config system (**replace the * with the ID of the joystick you want to use, a number between 1 and 16**):

- JOY_LX_*
- JOY_LY_*
- JOY_RX_*
- JOY_RY_*

Each is a axis. For example: JOY_LX_1 is the left/right of the left analog stick on controller 1, while JOY_LY_1 is the up/down axis of the same stick on controller 1.





JOY_RX_1 and JOY_RY_1 will be the left/right, up/down of the right stick on controller 1.

Each controller will implement its layout differently. But Orx makes use of the SDL_GameControllerDB community database internally to make the mapping consistent for all controllers playing your game

To allow the user to customise mapping, it is helpful to consider providing remapping in your game.

Stick Threshold

When pushing the stick in any direction, after getting past the small threshold, the value can be read using the `orxInput_IsActive` and `orxInput_GetValue` functions.

Let's work through setting this up. Firstly, `init` up a new project using these instructions.

Once you have a working project, add in the following graphic into your project's data to act as the object that will be controlled by the joystick:



Next, change the default Object in the config to use the `ball.png` graphic, and also to use a body so that we can have some physics (and deceleration) on the object:

```
[Object]
Graphic           = @
Texture           = ball.png
Pivot             = center
Position          = (0, 0, 0)
```

```
Body            = ObjectBody

[ObjectBody]
LinearDamping = 1.5
FixedRotation = true
Dynamic       = true
AllowSleep    = false
PartList      = ObjectBodyPart

[ObjectBodyPart]
Type  = box
Solid = true
```

Next, we will define the joystick direction controls in the `[MainInput]` section:

```
[Input]
SetList          = MainInput
DefaultThreshold = 0.2

[MainInput]
JOY_LX_1   = LeftRight
JOY_LY_1   = UpDown
KEY_ESCAPE = Quit
```

The `JOY_LX_1` above means, get values from the left analog stick when it is pushed along the X axis, either left or right. No value is read until it crosses the threshold (`DefaultThreshold`), which is 0.2, After that, register a `LeftRight` value.

The `_1` at the end of `JOY_LX_1` means joystick #1. There can be up to 16 physical USB inputs used.

In the same way, the `JOY_LY_1` means that if the analog stick has been moved up or down the Y axis, it will register an `UpDown` value.

For more details, see: Analog Stick Threshold

Finally in the `Run()` function, we can add the code to respond to these “movements”. `Run()` is not normally the recommended place to put this code, but for demonstration purposes, it is fine:

```
        const orxFLOAT speed = 150;

    if (orxInput_IsActive("LeftRight")) {
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);

        orxFLOAT x = orxInput_GetValue("LeftRight");
        speedVector.fX = x * speed;

        orxObject_SetSpeed(object, &speedVector);
    }

    if (orxInput_IsActive("UpDown")) {
```

```
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);

        orxFLOAT y = orxInput_GetValue("UpDown");
        speedVector.fY = y * -speed;

        orxObject_SetSpeed(object, &speedVector);
    }
```

In the above code, the `LeftRight` and `UpDown` inputs are being checked to see if they are active. If so, get the current speed and the axis position values with `orxInput_GetValue` and set the proper speed direction value to 150 / the axis the value. The smaller the stick movement, the smaller the object movement.

It's all pretty simple.

Further reading

Steam have an interesting article on their controller usage metrics, tools and experiments in controller remapping.

This is a different take to Orx's use of the SDL_GameControllerDB community database.

You can read the article on Steam here.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/input/analog_joystick_gamepad_control

Last update: **2025/09/30 17:26 (14 months ago)**