8 Way Joystick / Gamepad Control

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.

This gives you smooth multi-direction control. Orx can automatically limit the reading of the axis' to allow behaviour that is similar to that of old 8-way joysticks from the era of the Commodore 64, Amstrad, ZX, Amiga, etc.

Orx provides this with the “Half Axis” properties from your `Input` config.

Here's the list of the available joystick half axes for the input config system which allow you to easily use the axis like an 8-way old school joystick (**replace the * with the ID of the joystick you want to use, a number between 1 and 16**):

- +JOY_LX_* (Binds the left joystick's positive X half-axis to an input)
- -JOY_LX_* (Binds the left joystick's negative X half-axis to an input)
- +JOY_LY_* (Binds the left joystick's positive Y half-axis to an input)
- -JOY_LY_* (Binds the left joystick's negative Y half-axis to an input)
- +JOY_RX_* (Binds the right joystick's positive X half-axis to an input)
- -JOY_RX_* (Binds the right joystick's negative X half-axis to an input)
- +JOY_RY_* (Binds the right joystick's positive Y half-axis to an input)
- -JOY_RY_* (Binds the right joystick's negative Y half-axis to an input)

When pushing the stick in a direction, after getting past the threshold, the push will be “ON”. Before the threshold, it will be off, emulating a digital joystick or gamepad direction inputs (d-pad).

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 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 = 0.5

[MainInput]
+JOY_LX_1   = GoRight
-JOY_LX_1   = GoLeft
+JOY_LY_1   = GoUp
-JOY_LY_1   = GoDown
KEY_ESCAPE = Quit
```

The `+JOY_LX_1` above means, if the left analog stick is pushed along the X axis, in the positive direction, and after it has crossed the threshold, which is half the entire distance the stick can move (0.5), register a `GoRight` button press.

In the same way, the `-JOY_LY_1` means the if the Y axis has been pulled down with the left stick, and has passed the halfway (0.5) threshold, register a `GoDown` press.

Visit Analog Stick Threshold for more details on thresholds.

Finally in the `Run()` function, we can add the code to respond to these "presses". `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("GoRight"))
    {
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);
        speedVector.fX = speed;
        if (!orxInput_IsActive("GoUp") && !orxInput_IsActive("GoDown")) {
            speedVector.fY = 0;
        }

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

```
    if (orxInput_IsActive("GoLeft"))
    {
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);
        speedVector.fX = -speed;
        if (!orxInput_IsActive("GoUp") && !orxInput_IsActive("GoDown")) {
            speedVector.fY = 0;
        }

        orxObject_SetSpeed(object, &speedVector);
    }

    if (orxInput_IsActive("GoUp"))
    {
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);
        speedVector.fY = -speed;
        if (!orxInput_IsActive("GoLeft") && !orxInput_IsActive("GoRight")) {
            speedVector.fX = 0;
        }

        orxObject_SetSpeed(object, &speedVector);
    }

    if (orxInput_IsActive("GoDown"))
    {
        orxVECTOR speedVector = orxVECTOR_0;
        orxObject_GetSpeed(object, &speedVector);
        speedVector.fY = speed;
        if (!orxInput_IsActive("GoLeft") && !orxInput_IsActive("GoRight")) {
            speedVector.fX = 0;
        }

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

In the above code, all four direction inputs are being checked to see if they are active. If so, get the current speed and set the proper speed direction value to 150.

There is also a check in each condition to ensure that if a direction is not diagonal, the perpendicular axis speed is zero'ed out. That will stop any goofy movement, and make the object feel more like a proper 8-way joystick.

That's all there is to it. Enjoy.

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

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/input/8way_joystick_control?rev=1598877459

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