Testing Keyboard, Joystick and Mouse Inputs

This is a small routine to test for all types on inputs, and to output their names on to the screen. A good way to check if your usb gamepad works and what the values are.

Good also for keyboard testing to get the names of keypresses which you can use in your config files.

Start by creating a new Creating your own Orx-based Project using 'init'.

Open the project. We don't need an `Object` so remove or comment out this line from the Init() function:

```
orxObject_CreateFromConfig("Object");
```

Use the core clock in the Init() function to check regularly for input:

```
orxClock_Register(orxClock_Get(orxCLOCK_KZ_CORE), InputCheck, orxNULL,
orxMODULE_ID_MAIN, orxCLOCK_PRIORITY_NORMAL);
```



Note: `orxInput_GetActiveBinding` only collects an input for the specific frame. A slow custom clock will miss values especially if they are fast inputs like mouse wheel clicks. Therefore, we use the **core clock**.

And finally, the `InputCheck` function to get and display input:

```
void orxFASTCALL InputCheck(const orxCLOCK_INFO *_pstClockInfo, void
*_pstContext)
{
    orxINPUT_TYPE inputType;
    orxENUM buttonAxisOrKeyID;
    orxFLOAT value;
    orxINPUT_MODE mode = orxINPUT_MODE_FULL;

    if (orxInput_GetActiveBinding(&inputType, &buttonAxisOrKeyID, &value))
    {
        const orxSTRING pressedInputName =
orxInput_GetBindingName(inputType, buttonAxisOrKeyID, mode);
        orxLOG("Input: %s, Type: %d, ID: %d, Value %f", pressedInputName,
inputType, buttonAxisOrKeyID, value);
    }
}
```

`orxInput_GetActiveBinding` will populate the `inputType` and `buttonAxisOrKeyID` variables which can be used to get the input name using `orxInput_GetBindingName`. This is handy for logging or getting input names that can be used in the config. Typical output will be:

```
[21:15:25] [LOG] Input: MOUSE_X, Type: 2, ID: 0, Value 771.000000
[21:15:26] [LOG] Input: KEY_H, Type: 0, ID: 7, Value 1.000000
[21:15:26] [LOG] Input: KEY_H, Type: 0, ID: 7, Value 1.000000
[21:15:27] [LOG] Input: KEY_H, Type: 0, ID: 7, Value 1.000000
[21:15:28] [LOG] Input: MOUSE_LEFT, Type: 1, ID: 0, Value 1.000000
[21:16:01] [LOG] Input: KEY_N, Type: 0, ID: 13, Value 1.000000
[21:16:01] [LOG] Input: KEY_SPACE, Type: 0, ID: 71, Value 1.000000
[21:16:01] [LOG] Input: KEY_SPACE, Type: 0, ID: 71, Value 1.000000
[21:16:02] [LOG] Input: JOY_LX_1, Type: 4, ID: 0, Value -0.447379
[21:16:02] [LOG] Input: JOY_LX_1, Type: 4, ID: 0, Value -0.833341
[21:16:04] [LOG] Input: JOY_1_1, Type: 3, ID: 0, Value 1.000000
[21:16:05] [LOG] Input: JOY_1_1, Type: 3, ID: 0, Value 1.000000
[21:17:01] [LOG] Input: JOY_LX_1, Type: 4, ID: 0, Value 0.300008
[21:17:01] [LOG] Input: JOY_LY_1, Type: 4, ID: 1, Value -0.454551
[21:17:02] [LOG] Input: JOY_16_1, Type: 3, ID: 15, Value 1.000000
```

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

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/input/testing_input_controls?rev=1670362762

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