Mouse Grab

You might notice on emulators and some games that use mouse movement, that the mouse is locked into the window or play area. The movement while contrained within the game window but the movement is able to continue infinitely in any direction.

You can then usually press a predefined key combo to free it from the application window so that items on the desktop can be accessed again.

Your player will get annoyed after a while and give the game away.

Orx lets you perform mouse grabbing and mouse releasing.

Let's try this out on a standard project. First, make your project.

Setting up a new project

To help you work through this tutorial, first create a new blank project using the init script.

Windowify the project

To demo this feature, we will change it from full screen to a smaller window, and make key presses that both grab and release the mouse.

First, change the config to make the full screen into a window, change the `Decorations` property under the `[Display]` section:

```
[Display]
...
Decoration      = false
```

Then specify a screen size. The entire `[Display]` section should look like this:

```
[Display]
Title           = mousegrab
FullScreen      = false
Decoration      = true
Smoothing       = true
VSync           = true
ScreenWidth     = 800
ScreenHeight    = 600
```

Compile and run. You'll see a demo application with the spinning Orx logo in a window.

Setting up rotation based on the mouse

Let's set up rotation of the Orx Logo based on the movement of the mouse.

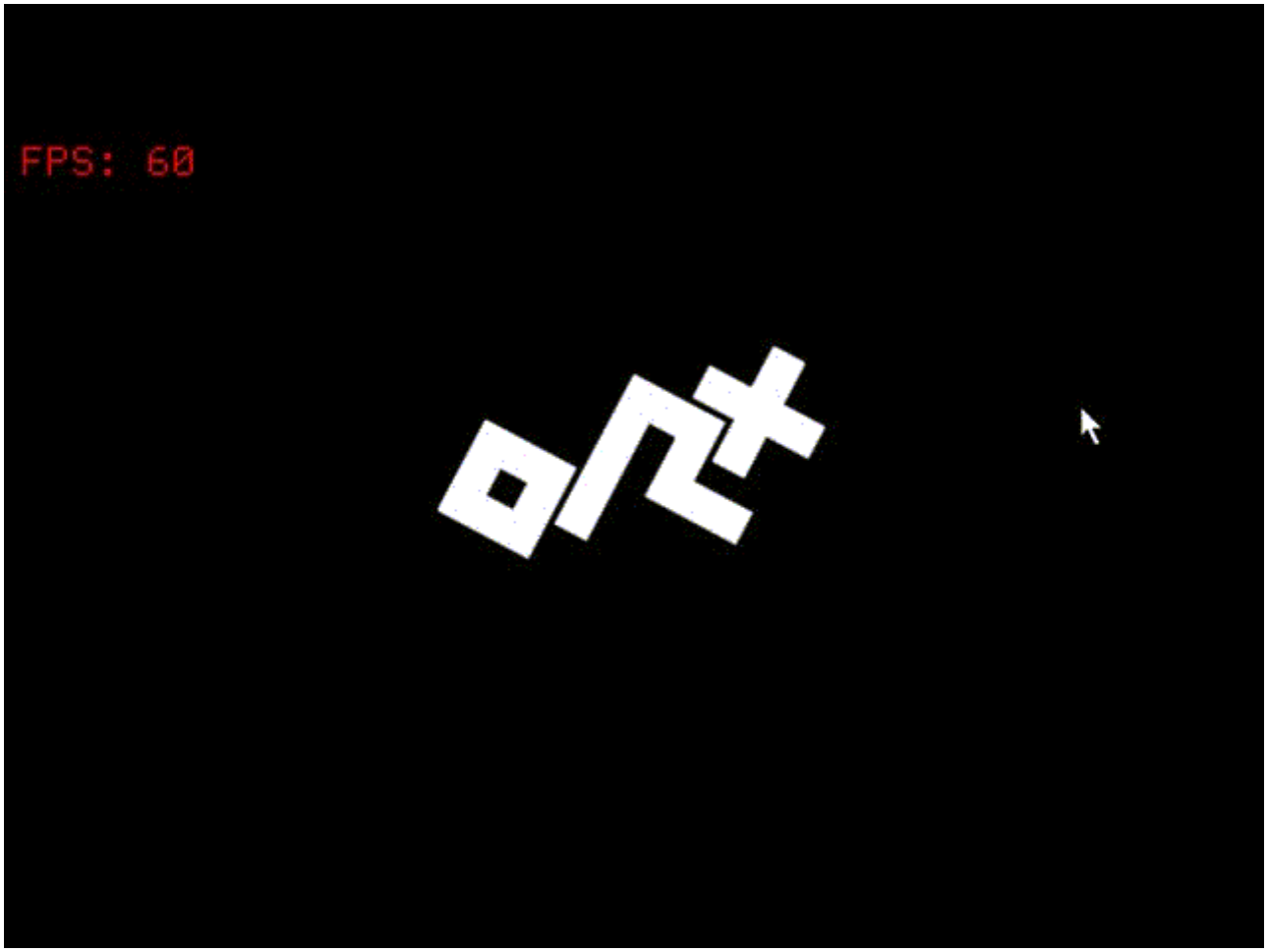
We can do this with a couple of commands in a Track which will be attached to the default Object:

```
[ObjectMouseRotationTrack]
0    = >Mouse.GetPosition # >Vector.X < # Object.SetRotation ^ <
Loop = true
```

And add the track to the Object:

```
[Object]
Graphic          = @
Texture          = logo.png
Pivot            = center
TrackList        = ObjectMouseRotationTrack
```

Run this and move the mouse left and right. Notice the spinning of the logo. But you can only go so far. The borders of the physical screen space is your boundary. And you can keep moving indefinitely in either direction. The screen edge will block you.



Mouse grab can overcome this limitation.

Testing the mouse grab with a command

Press the ~ key (tilde) to open the Orx Console while the application is running.

```
Mouse.Grab true
```

Press the ~ key (tilde) to close the Orx Console, which returns to the running application.

Notice the mouse has completely disappeared. But you are now able to continually move in either direction indefinitely and spin the logo indefinitely too.

Press the ~ key (tilde) to re-open the Orx Console.

```
Mouse.Grab false
```

The mouse should reappear again.

Press the ~ key (tilde) to close the Orx Console, and press `Esc` to quit the game.

Implementing in Code

Let's define a key press to be able to toggle the grabbing of the mouse.

```
[Input]
KEY_ESCAPE      = Quit
KEY_M      = ToggleMouseGrabbing
```

To keep track of the grab state, add a variable to the top of the code file:

```
#include "orx.h"

orxBOOL mouseGrabbed = orxFALSE;
```

And finally, add the following to the bottom of the `Update()` function:

```
    if(orxInput_HasBeenActivated("ToggleMouseGrabbing"))
    {
        mouseGrabbed = !mouseGrabbed;
        orxMouse_Grab(mouseGrabbed);
    }
```

Compile and run, and move the mouse left and right to confirm you can only travel as far as the screen boundaries. Press the `M` key to grab the mouse and test that you can move the mouse indefinitely in either direction.

Press `M` again to release the grab. The mouse cursor will reappear and you are again constrained by the screen boundaries.

Pretty neat!

Note: The `ShowCursor` property of the `[Mouse]` section is ignored if grab is true.