whatido

WhatIDo is a little tool that shows what you do with your Keyboard and Mouse.

The aim of this tool is the project JNativehook and java FX 8.

Only tested on Windows and Linux (Debian) , AZERTY Keyboard, and with a Java VM Oracle 8 (not tested with a QWERTY keyboard)

The tool is almost of the time in top of all the windows (except for contextual Windows Menus that are prioritary)

The archive **whatidoExe.zip,** contains the executable . It must be dezipped in a folder, and under the folder script, you have to configure the scripts to launch your application.

Example for Windows in red color , the variables to adapt (to be done with the four files of the script folder) :

```
Set PROJECT_HOME=C:\opt\whatido
Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar;.;%PROJECT_HOME
%\lib\whatido-1.0.0.jar;
Set JAVA_HOME=C:\Program Files (x86)\Java\jre1.8.0_121\bin
start "" "%JAVA_HOME%\javaw" -Droot=%PROJECT_HOME% -Dhome=
%PROJECT_HOME% -cp %CLASSPATH% com.jlp.whatido.Main
Exit
```

Other important configurations, (that are supposed correct for that I have tested) are located in the **config/keyboards** folder and contains the mapping with the characters of the keyboard and the **NativeEvent.getKeyCode() of the JnativeHook API**. For exemple my desktop Debian AZERTY Keyboard gives QWERTY native code, so I re-map with the correspondant files.

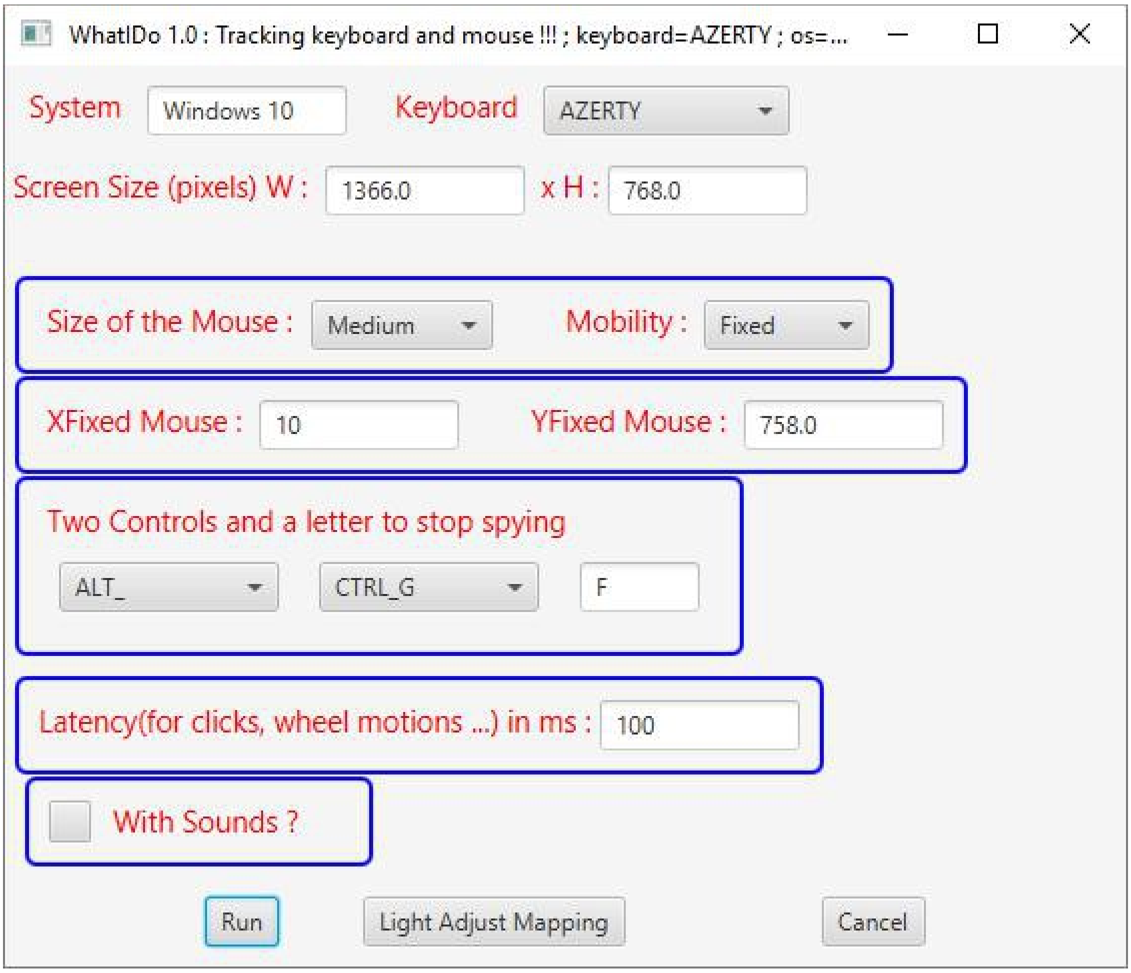
These files are key/value (Java Properties Class) pair and important : **the key must be unique and also the value must be unique. When a keyboard key has several character, you must chosee only one.**

To modify the mapping, you can use the script : **testingMouseKeyBoard** to retrieve all the code and correct the mapping.

You can also modify with the tool, if there are few keys to modify.

Some screen shots :

To configure OS / Screen/ Keyboard/ Size of the mouse / Mouse Fixe or Mobile/distance from the device pointer when Mobile/ Stop Combinaisons
/ Latency of the messages/ With sound spelling or not …



WhatIDo 1.0 : Tracking keyboard and mouse !!! ; keyboard=AZERTY ; os=...

System Windows 10 Keyboard AZERTY

Screen Size (pixels) W : 1366.0 x H : 768.0

Size of the Mouse : Medium Mobility : Fixed

XFixed Mouse : 10 YFixed Mouse : 758.0

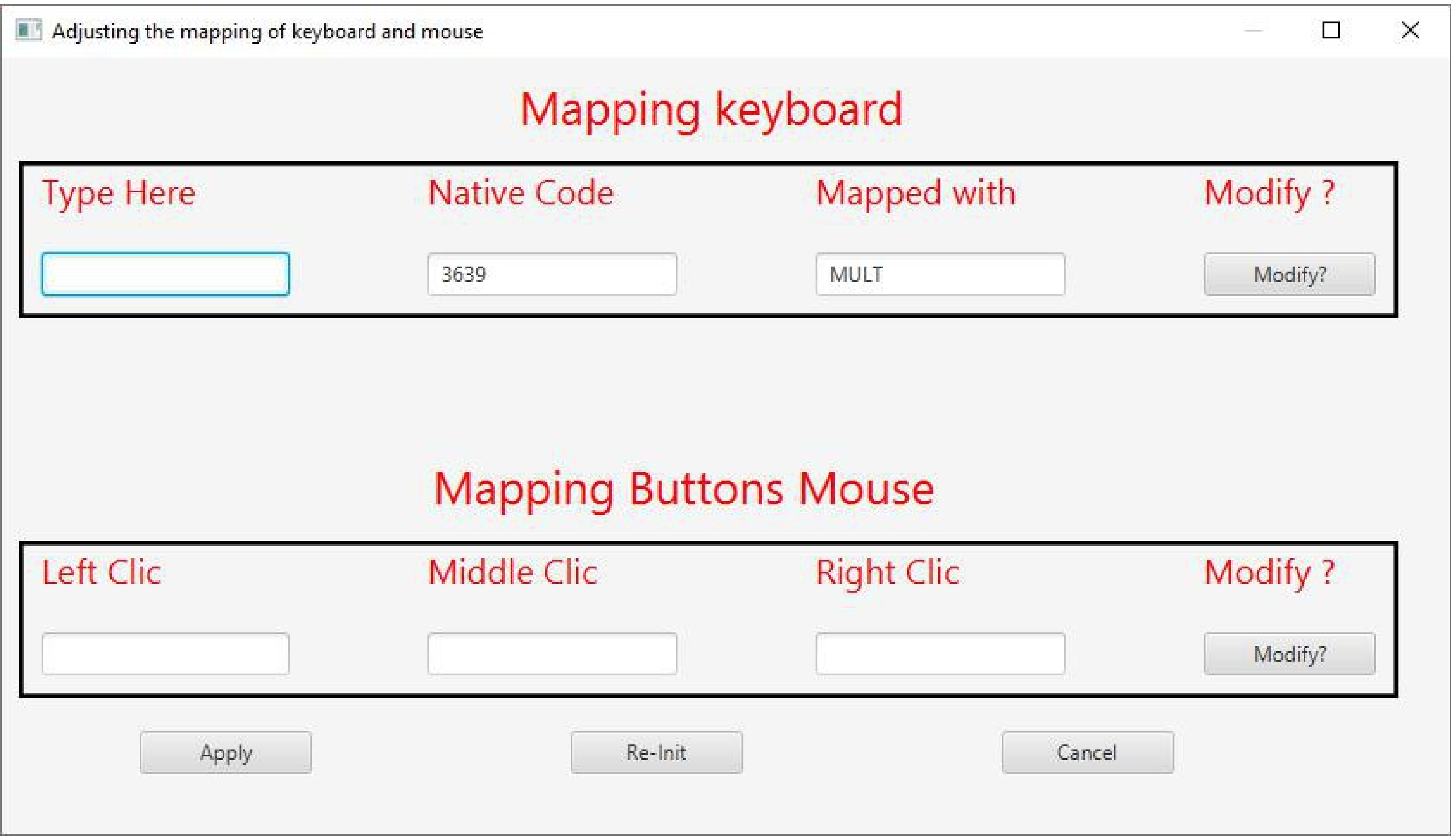
Two Controls and a letter to stop spying

ALT_ CTRL_G F

Latency(for clicks, wheel motions ...) in ms : 100

□ With Sounds ?

A possibility for correct the mapping (if there are few keys to modify) :



Adjusting the mapping of keyboard and mouse

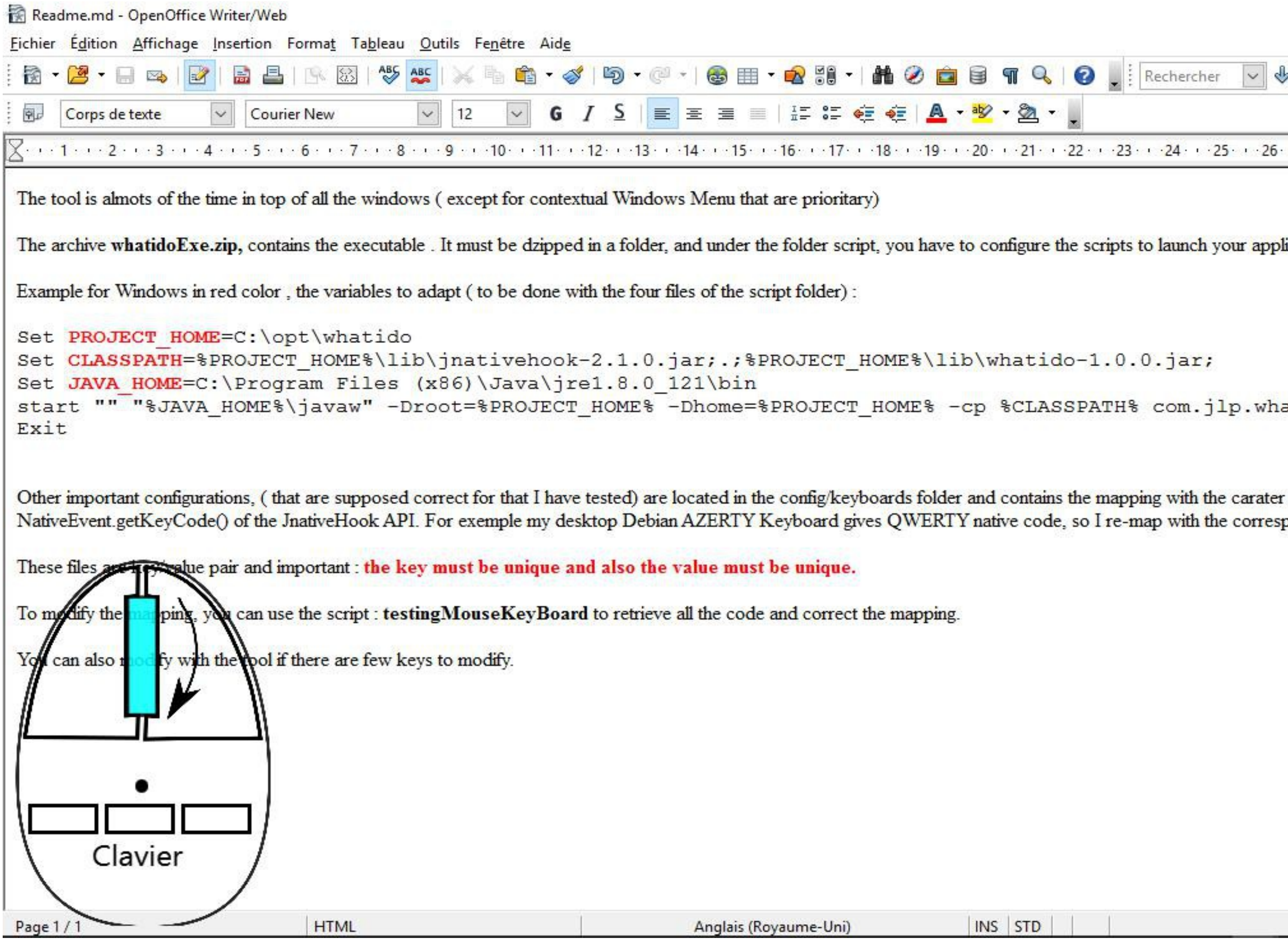
Mapping keyboard

Type Here	Native Code	Mapped with	Modify ?
	3639	MULT	

Mapping Buttons Mouse

Left Clic	Middle Clic	Right Clic	Modify ?

And in action, when I turn the wheel :


Readme.md - OpenOffice Writer/Web
Fichier Édition Affichage Insertion Format Tableau Outils Fenêtre Aide
Rechercher
Corps de texte
Courier New
12
G I S
Clavier
Page 1 / 1
HTML
Anglais (Royaume-Uni)
INS STD

The tool is almots of the time in top of all the windows (except for contextual Windows Menu that are prioritary)

The archive **whatidoExe.zip,** contains the executable . It must be dzipped in a folder, and under the folder script, you have to configure the scripts to launch your appli

Example for Windows in red color , the variables to adapt (to be done with the four files of the script folder) :

```
Set PROJECT_HOME=C:\opt\whatido
Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar;.;%PROJECT_HOME%\lib\whatido-1.0.0.jar;
Set JAVA_HOME=C:\Program Files (x86)\Java\jre1.8.0_121\bin
start "" "%JAVA_HOME%\javaw" -Droot=%PROJECT_HOME% -Dhome=%PROJECT_HOME% -cp %CLASSPATH% com.jlp.wha
Exit
```

Other important configurations, (that are supposed correct for that I have tested) are located in the config/keyboards folder and contains the mapping with the carater NativeEvent.getKeyCode() of the JnativeHook API. For exemple my desktop Debian AZERTY Keyboard gives QWERTY native code, so I re-map with the corresp

These files are key/value pair and important : **the key must be unique and also the value must be unique.**

To modify the mapping, you can use the script : **testingMouseKeyBoard** to retrieve all the code and correct the mapping.

You can also modify with the tool if there are few keys to modify.