

MANUEL UTILISATEUR

WHATIDO

Version 3.0

Date	Version	Commentaire
10/04/17	V3.0	Version initiale du document
21/04/17	V3.0.1	Ajout de 2 tailles de clavier

Table des matières

1 Présentation du produit.....	3
1.1 Généralités.....	3
1.2 Un aperçu du produit sur ce document.....	3
1.3 Exemples d'utilisation du produit.....	4
2 Installation.....	5
2.1 Pré-requis.....	5
2.2 Installation de Whatido.....	6
3 Manuel utilisateur.....	8
3.1 Configuration du clavier pour les démonstrations.....	8
3.2 Écran de configuration initial.....	9
3.3 Souris Fixed (Large, Medium,Small) seule.....	10
3.4 Souris Mobile (Medium ,Small, Tiny) seule.....	11
3.5 Activation de la visibilité du clavier.....	12
4 Annexe.....	15
4.1 Préambule.....	15
4.2 Le mapping du clavier.....	16
4.2.1 Donner un identifiant au clavier.....	16
4.2.2 Les fichiers à créer pour le mapping clavier :.....	17
4.2.2.1 config/keyboards/OCCITAN_Code.properties.....	17
4.2.2.2 config/keyboards/OCCITAN_CtrlMod.properties.....	18
4.2.2.3 config/keyboards/OC_oc_Trad.properties.....	19
4.3 La visualisation du clavier.....	20
4.3.1 Image png du clavier OCCITAN.....	20
4.3.2 Localisation des touches dans l'image.....	20
4.4 Mettre du Son.....	21
4.4.1 Fichiers de configurations	22

1 Présentation du produit

1.1 Généralités

Il s'agit d'un utilitaire qui permet de visualiser sur l'écran, les actions que l'on fait sur la souris (clic / droit/gauche/milieu et les actions avant / arrière sur la roulette) ainsi que les actions sur le clavier (hors pavé numérique).

La visualisation se fait à travers 2 images transparentes de la souris et du clavier qui reste en fenêtre toujours visible (sauf pour quelques menus contextuel où Windows est prioritaire, mais ce n'est pas très gênant).

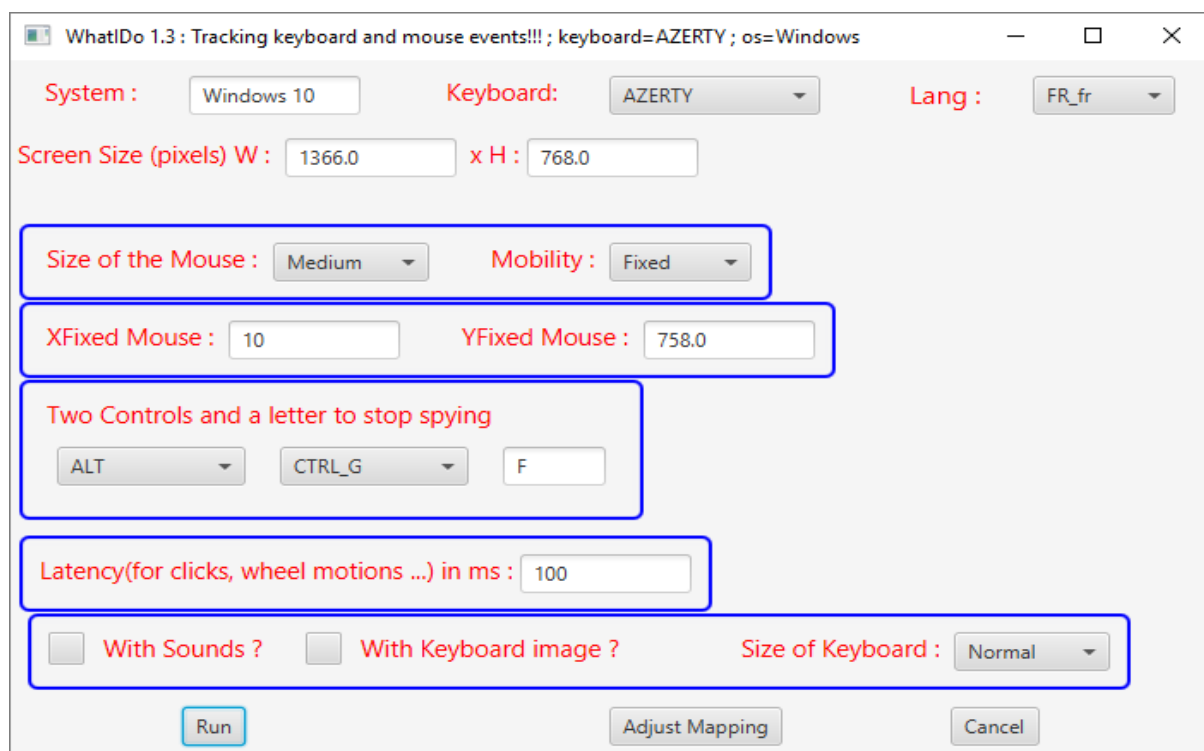
Le logiciel fonctionne aussi sur Linux et à priori aussi sur OS X (non testé par moi sur OS X, mapping du clavier à faire), voir dans les annexes de ce document comment réaliser un mapping complet si ce logiciel ne fonctionne pas correctement avec votre micro/os/clavier. Il se peut qu'il y ait des écarts avec le mapping Windows/AZERTY .

Cette application est basée sur la Bibliothèque **JNativeHook** disponible sur le site Github : <https://github.com/kwhat/jnativehook>

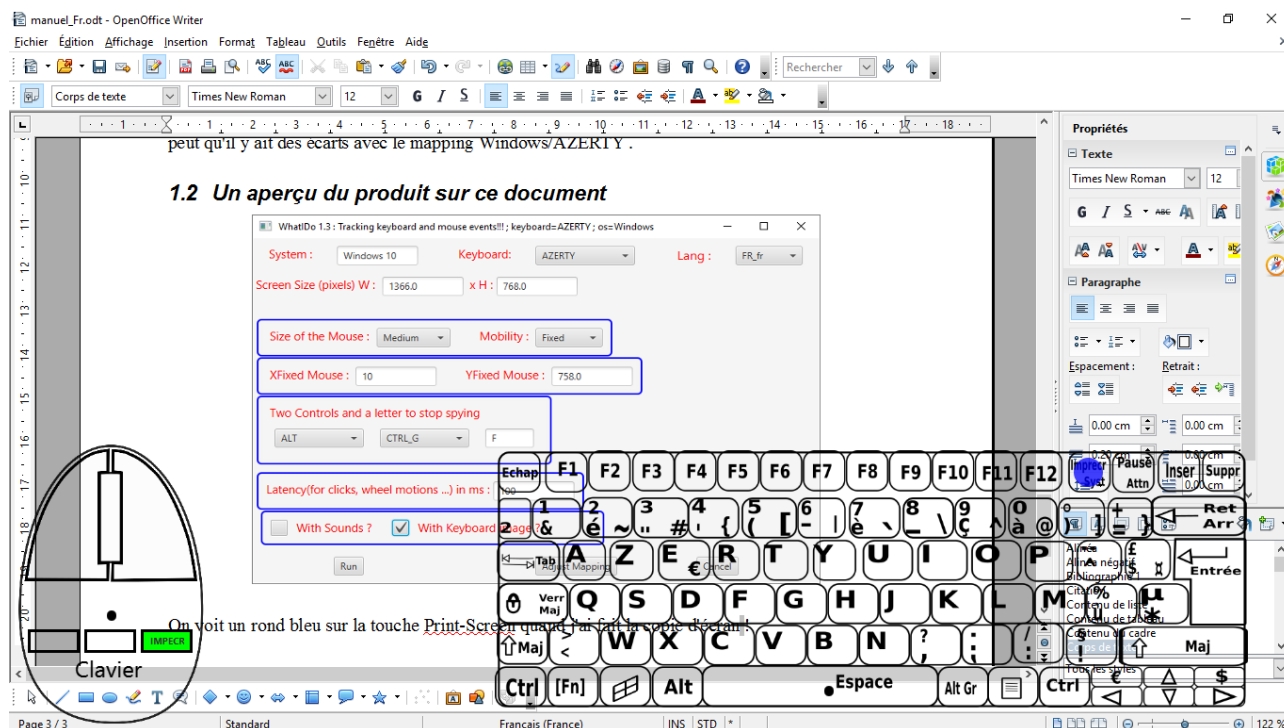
Le code binaire est disponible dans le dépôt Maven , j'ai utilisé la Version 2.1.0 : <https://mvnrepository.com/artifact/com.1stleg/jnativehook/2.1.0>

1.2 Un aperçu du produit sur ce document

L'écran de lancement qui sera expliqué en détail plus loin dans le document.



Une image du produit en action :



On voit un rond bleu sur la touche Print-Screen quand j'ai fait la copie d'écran !

On peut choisir l'affichage ou non du clavier, on peut activer le son qui décrit les actions faites sur la souris et le clavier.

La taille de l'image du clavier peut être réduite de moitié ou au $\frac{3}{4}$ en choisissant dans la combobox (Normal, Medium, Small)

Pour ce qui concerne la souris, on a 4 tailles au choix (Large, Medium, Small, Tiny) et le fait d'être mobile pour les 3 plus petites tailles et d'être fixe pour les 3 plus grandes tailles.

Pour les souris de type fixe, le bouton noir au milieu permet de la déplacer dans une autre partie de l'écran quand elle gêne.

1.3 Exemples d'utilisation du produit

Whatido fonctionne correctement avec des vidéo-projecteurs.

Les utilisations possibles sont :

- initiation à l'informatique pour la présentation des actions souris et le clavier
- présentation en direct de logiciel en visualisant toutes les actions souris et clavier
 - ex : navigation dans l'explorateur de fichier, mécanismes du copier/coller, utilisation des outils bureautiques ...
- création de tutoriels vidéo par enregistrement de l'écran avec **whatido** activé.

Astuce : on peut lancer 2 fois le produit :

- une fois avec une souris fixe de taille Large ou Médium
- l'autre fois avec la souris Tiny/Mobile qui va suivre le curseur Windows au cours de ses déplacements.

Les 2 souris montreront les actions faites.

Limitation avec **Powerpoint** en mode Diaporama, le produit (image souris et/ou image clavier) n'apparaît pas en premier plan et ne peut être utilisé dans ce cas, il faut rester en mode édition.

2 Installation

2.1 Pré-requis

Whatido nécessite la présence d'une machine virtuelle **Java** récente version supérieure à **1.8.0_121**.

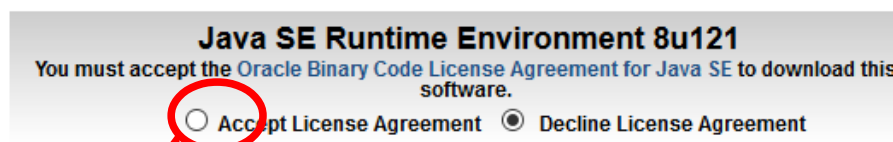
On pourra installer la version **JRE 32 bits** depuis le site de **Oracle** :

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

JRE 8u121 Checksum



Cocher l'acceptation de licence.

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

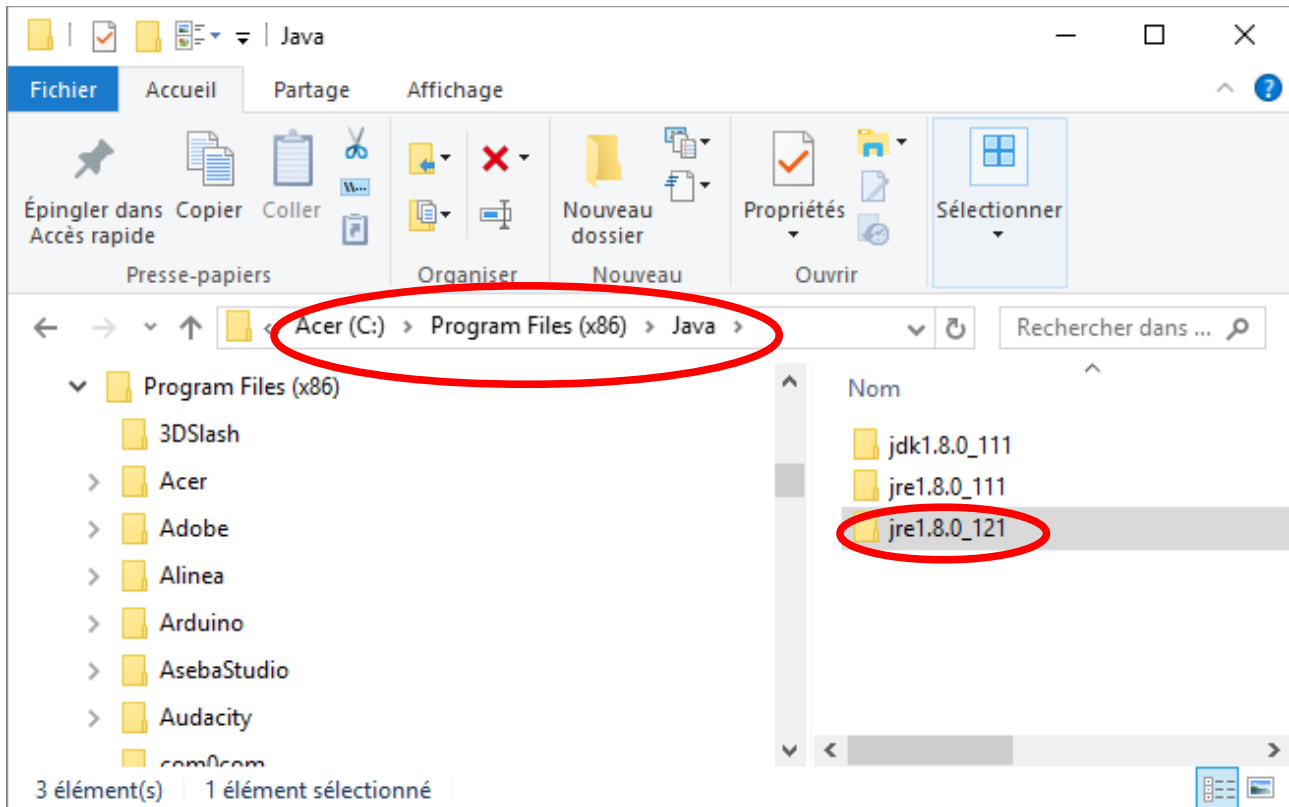
JRE 8u121 Checksum

Java SE Runtime Environment 8u121			
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.			
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you now download this software.			
Product / File Description	File Size	Download	
Linux x86	56.92 MB	↓	jre-8u121-linux-i586.rpm
Linux x86	72.76 MB	↓	jre-8u121-linux-i586.tar.gz
Linux x64	54.39 MB	↓	jre-8u121-linux-x64.rpm
Linux x64	70.26 MB	↓	jre-8u121-linux-x64.tar.gz
Mac OS X	62.28 MB	↓	jre-8u121-macosx-x64.dmg
Mac OS X	53.91 MB	↓	jre-8u121-macosx-x64.tar.gz
Solaris SPARC 64-bit	52.05 MB	↓	jre-8u121-solaris-sparcv9.tar.gz
Solaris x64	49.9 MB	↓	jre-8u121-solaris-x64.tar.gz
Windows x86 Online	0.7 MB	↓	jre-8u121-windows-i586-ifw.exe
Windows x86 Offline	53.81 MB	↓	jre-8u121-windows-i586.exe
Windows x86	59.17 MB	↓	jre-8u121-windows-i586.tar.gz
Windows x64 Offline	61.18 MB	↓	jre-8u121-windows-x64.exe
Windows x64	62.66 MB	↓	jre-8u121-windows-x64.tar.gz

Choisir le téléchargement de la version 32 bits (i586) . La version disponible à la rédaction du document est la version 8u121, prendre la dernière version disponible.

Après téléchargement, il faudra lancer le **fichier .exe correspondant en mode administrateur** (clic droit sur le fichier, et choisir lancement en mode administrateur).

Une fois installé, vous devez pouvoir voir votre JRE comme montré ci-dessous :



2.2 Installation de Whatido

Le produit se présente sous la forme d'une archive zip **whatido<Version>Exe.zip** ex **whatido3Exe.zip** .

On va supposer pour la suite de l'installation qu'il existe un répertoire **C:\opt** sur votre ordinateur, sinon vous le créez ou bien vous adaptez la procédure décrite ci-dessous.

- Positionner l'archive **whatido3Exe.zip** sous **C:\opt** et dézipper
- Ouvrir le fichier C:\opt\whatido\script\whatido.cmd et adapter les chemins en rouge

```
Set PROJECT_HOME=C:\opt\whatido
Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar;.;%PROJECT_HOME%\lib\whatido-3.0.0.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
```

- Ouvrir le fichier C:\opt\whatido\script\testingMouseKeyBoard.cmd et adapter les chemins en rouge

```
Set PROJECT_HOME=C:\opt\whatido
Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar;.;%PROJECT_HOME%\lib\whatido-3.0.0.jar;
Set JAVA_HOME=C:\Program Files (x86)\Java\jre1.8.0_121\bin
"%JAVA_HOME%\java" -Droot=%PROJECT_HOME% -Dhome=%PROJECT_HOME% -cp %CLASSPATH% com.jlp.whatido.MyMouseKeyboardListener
```

- Faire un raccourci sur le bureau pour le fichier C:\opt\whatido\script**whatido.cmd**
- Cliquer sur le raccourci pour lancer **whatido**.

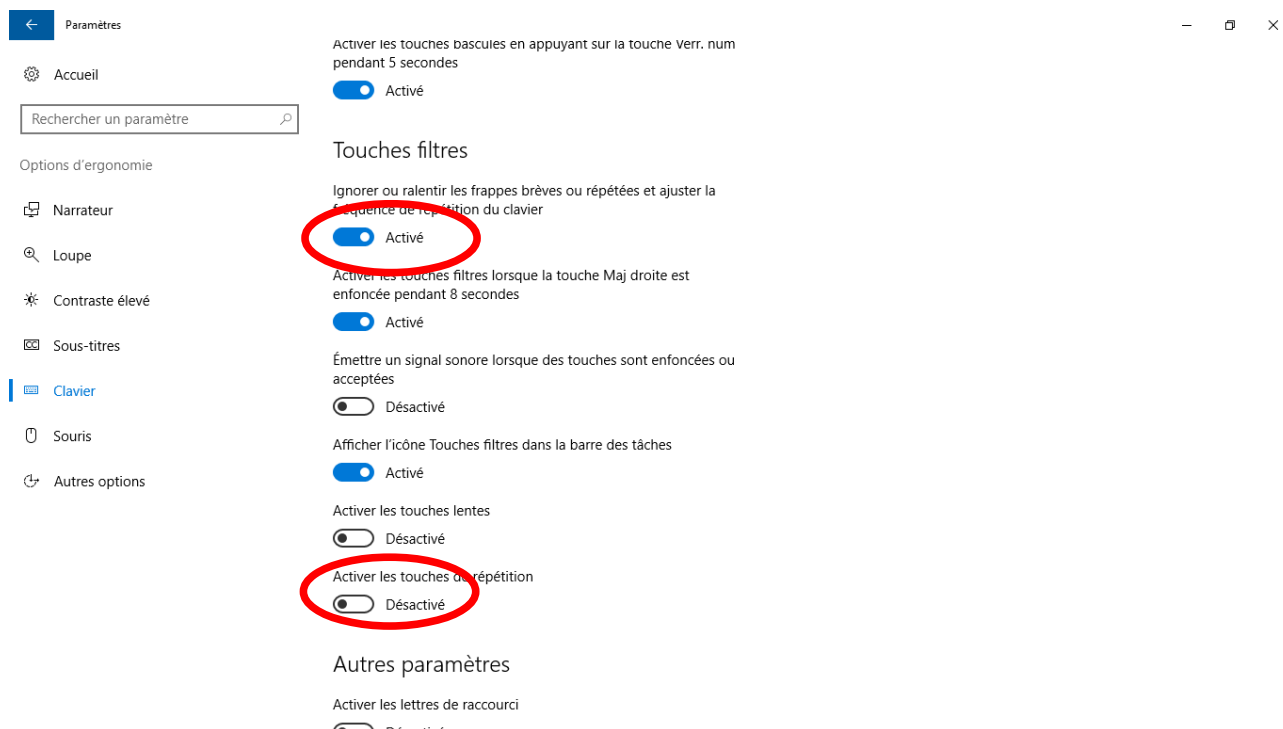
3 Manuel utilisateur

3.1 Configuration du clavier pour les démonstrations

Afin d'éviter la répétition non voulue des touches claviers, lorsque l'on garde la touche appuyée, il faut procéder aux réglages suivants pour Windows 10 (il existe des mécanismes identiques pour les autres versions de Windows ainsi que les différents types de Windowing Linux : KDE, XFCE ...) :

Paramètres → Option d'ergonomie → Clavier → Activer les Touches Filtre.

La configuration doit être conforme à l'écran ci-dessous :



Il faut activer les touches filtres et désactiver la répétition des touches.

3.2 Écran de configuration initial

Ci dessous cet écran avec les explications pour chaque objet numéroté dans l'image

WhatIDo 1.3 : Tracking keyboard and mouse events!!!; keyboard=AZERTY ; os=Windows

System : Windows 10 **1** Keyboard: AZERTY **2** Lang : FR_fr **3**

Screen Size (pixels) W : 1366.0 x H : 768.0 **4**

Size of the Mouse : Medium **5** Mobility : Fixed **6**

XFixed Mouse : 10 **7** YFixed Mouse : 758.0

Two Controls and a letter to stop spying **8**

ALT CTRL_G F

Latency(for clicks, wheel motions ...) in ms : 100 **9**

☒ With Sounds ? **10** ☐ With Keyboard image ? **11** Size of Keyboard : Normal

Run **14** Adjust Mapping **13** Cancel **12**

1 : Détection automatique du Système d'exploitation, rien à rentrer

2 : Choix du clavier parmi une liste. Lié à un mapping différent en fonction Clavier/OS. Choix à faire. On peut rajouter des claviers si nécessaire, voir en annexe.

3 : Choix de la langue : surtout utilisé quand le son est activé. Disponible en Français et Anglais. Voir en annexe comment étendre à d'autres langues.

4 : Détection automatique de la taille de l'écran (yc la taille disponible avec un vidéo-projecteur). Ne rien modifier

5 : Choix à faire sur la taille de la souris (Large, Medium, Small, Tiny)

6 : Choix du comportement de la souris **Fixed** (immobile mais déplaçable) , ou **Mobile** (suit le curseur système).

Large ne peut être que Fixed, Tiny ne peut être que Mobile. Medium et Small ont les 2 possibilités.

7 : Dans le cas **Fixed**, positionne la localisation de la souris au lancement de **whatido**.

Dans le cas **Mobile** on a :



DistX. Cursor (px) : DistY. Cursor (px) :

ce qui permet de fixer la distance en pixel entre le curseur système et l'image de la souris

8: Combinaison de touche permettant l'arrêt de **whatido** (2 contrôles et un caractère ex : ALT CTRL_G F). On peut aussi arrêter l'application en fermant la fenêtre de l'icône Java dans la barre des tâches.

9 : Pour améliorer la visibilité des actions, permet de régler la latence (en ms) des images de la souris et du clavier après relâchement du bouton de la souris ou de la touche du clavier.

10 : Possibilité de mettre le son qui décrit les actions faites sur la souris et le clavier (vite lassant!)

11 : Activation de la visualisation du clavier en bas à droite de l'écran. Un bouton noir au milieu de la barre espace permet de le positionner ailleurs dans l'écran. On peut choisir la taille de l'image du clavier dans la combobox.

12 : Sortir sans lancer l'application

13 : bouton de mapping clavier et souris Explications données en annexe, utilisation avancée.

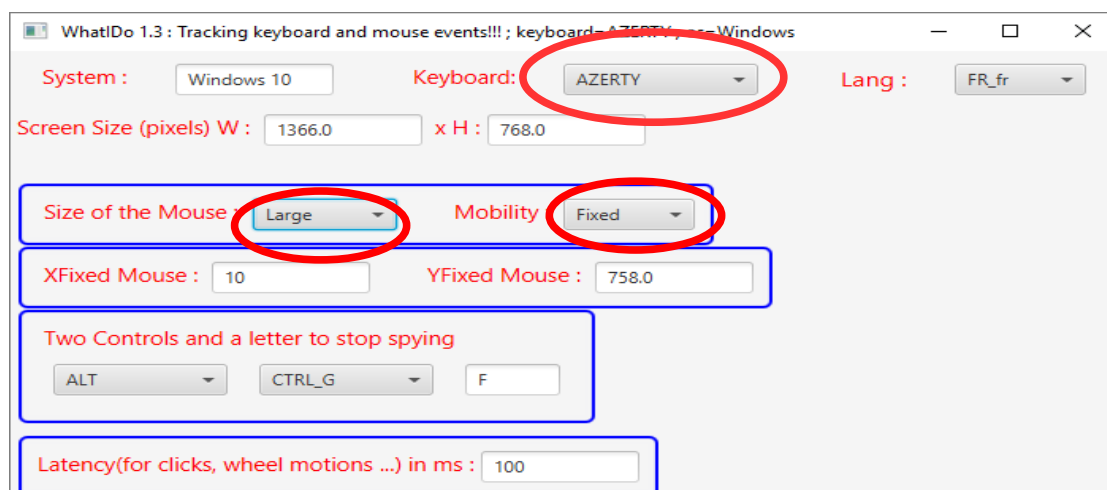
14 : bouton de lancement de l'application.

3.3 Souris Fixed (Large, Medium, Small) seule.

La procédure est décrite pour la souris de taille Large, mais elle est identique pour les 2 autres tailles Medium et Small.

Le fait d'activer ou pas le son n'a pas d'impact dans le fonctionnement décrit ci-dessous.

L'écran de configuration initial a donc l'aspect suivant :



WhatIDo 1.3 : Tracking keyboard and mouse events!!! ; keyboard= AZERTY, --Windows

System : Windows 10 Keyboard: AZERTY Lang : FR_fr

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

Size of the Mouse : Large Mobility : Fixed

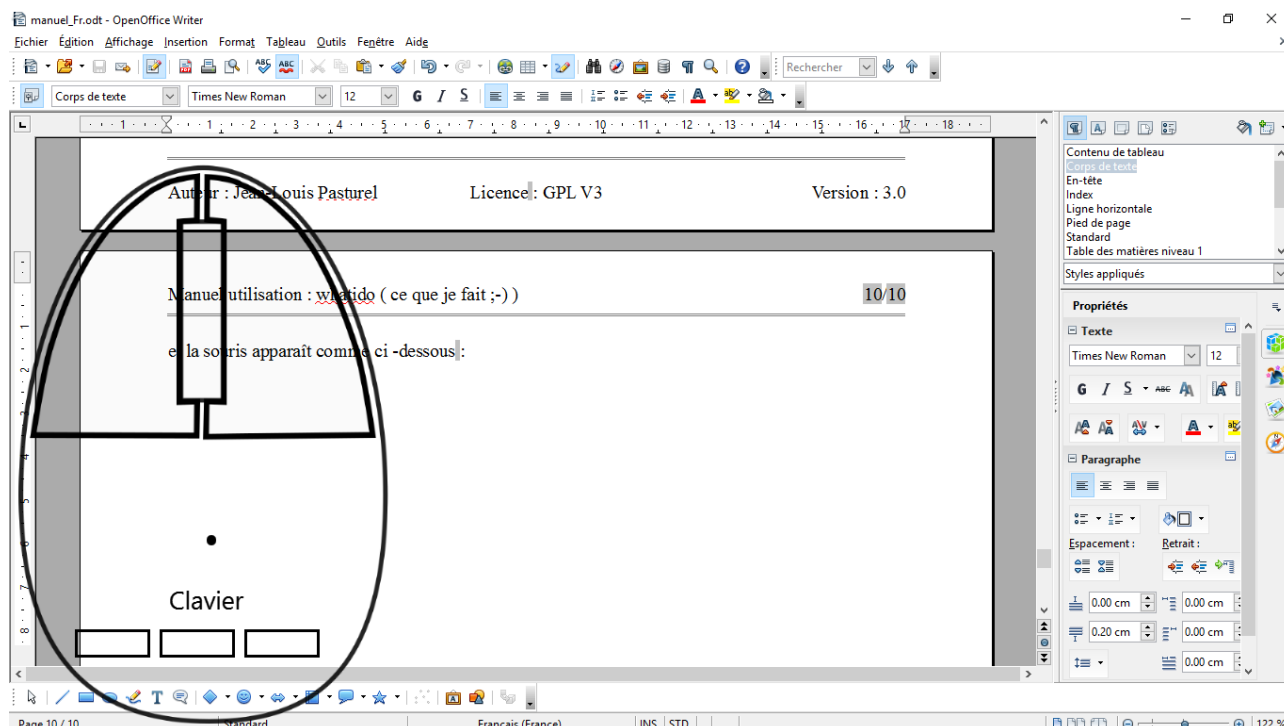
XFixed Mouse : 10 YFixed Mouse : 758.0

Two Controls and a letter to stop spying

ALT CTRL_G F

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

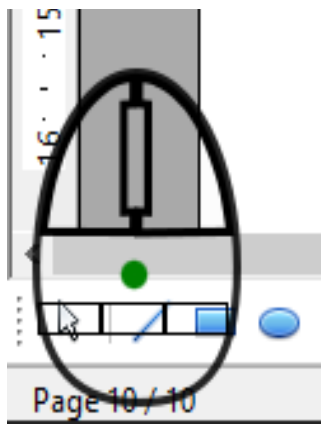
et la souris apparaît comme ci -dessous :



Chaque action sur la souris indique le bouton pressé (bleu pour bouton gauche, rouge pour bouton droit et central , déplacement roulette haut en rouge et bas en bleu)

Les 3 zones de textes de la souris indiquent les contrôles saisis (1ier et deuxième), le troisième indique le caractère du clavier.

Si on clique sur le bouton noir de la souris, il devient Vert. Si on clique ailleurs dans l'écran, la souris va se relocaliser dans la région cliquée.

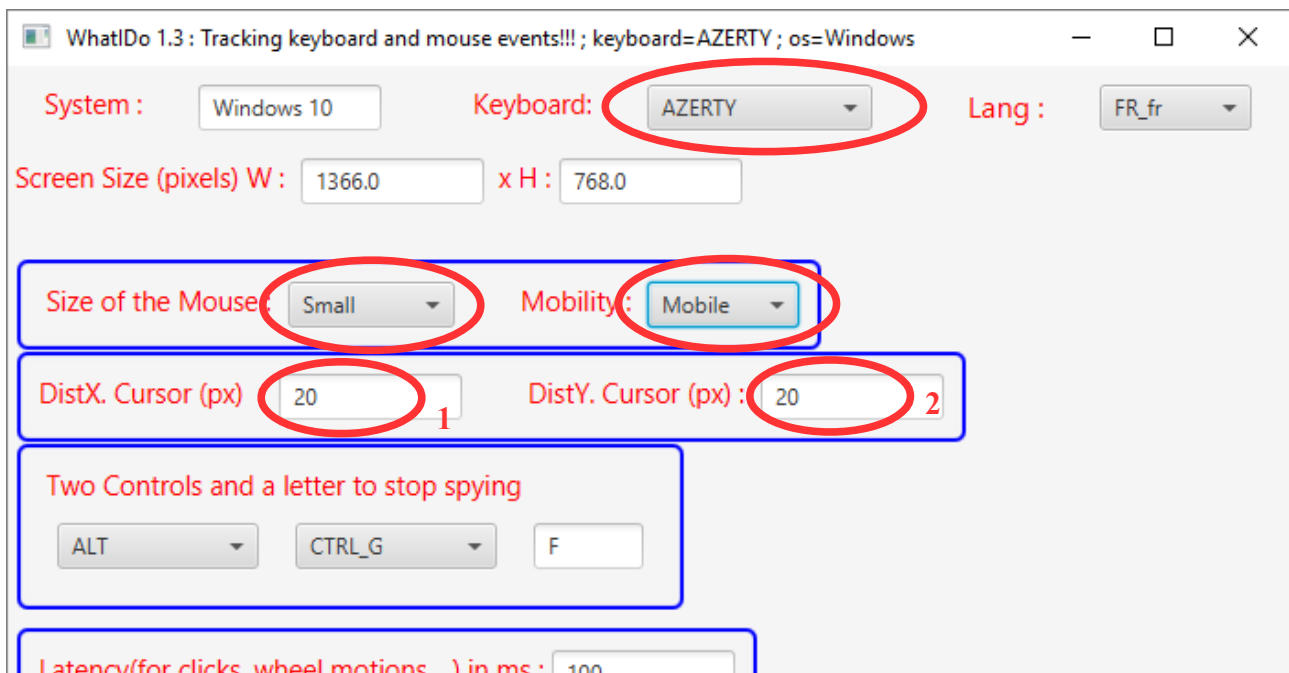


3.4 Souris Mobile (Medium ,Small, Tiny) seule

La procédure est décrite pour la souris de taille Small, mais elle est identique pour les 2 autres tailles Medium et Tiny .

Le fait d'activer ou pas le son n'a pas d'impact dans le fonctionnement décrit ci-dessous.

L'écran de configuration initial a donc l'aspect suivant :

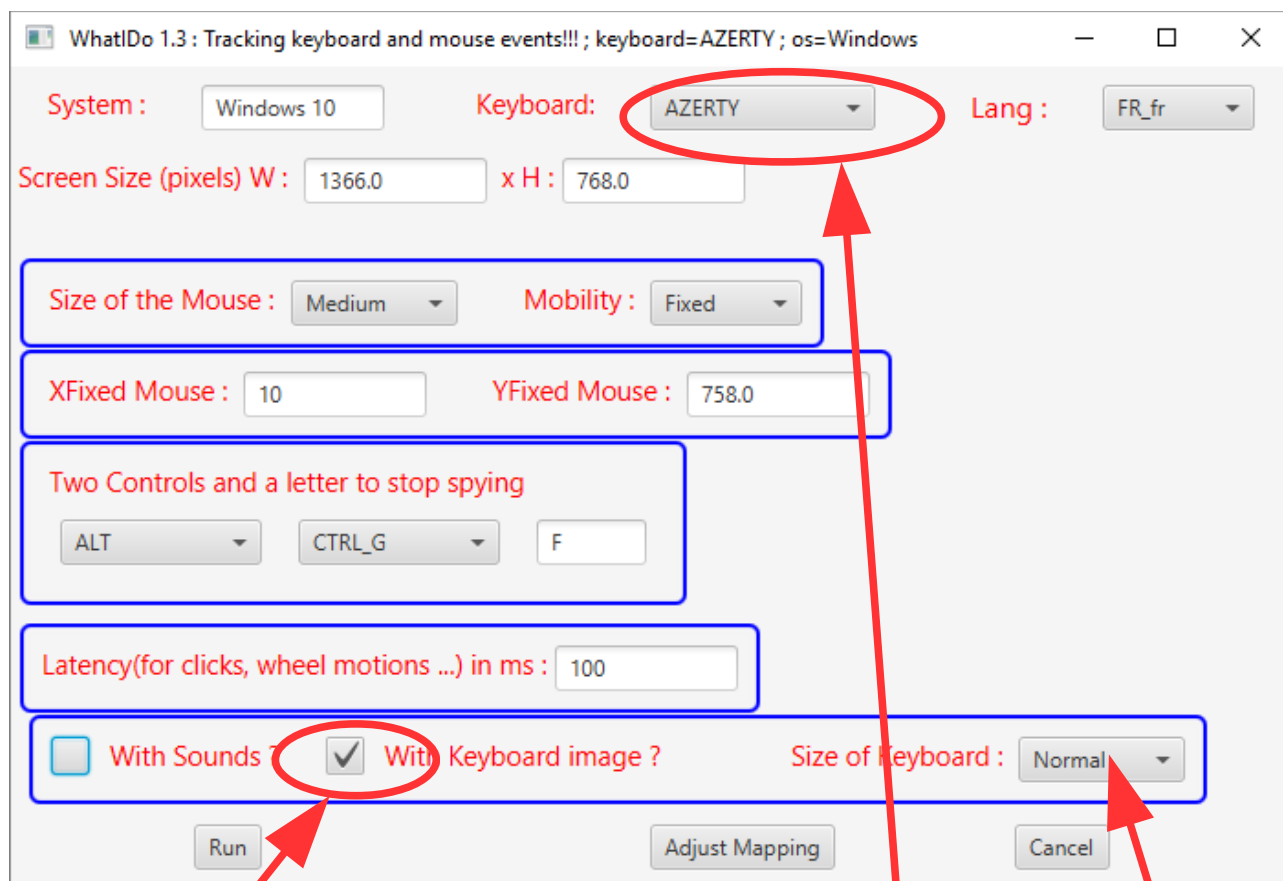


Après le lancement, l'image de la souris va suivre le curseur système lors de ses déplacements. La distance est réglée par les paramètres identifiés 1 et 2 dans l'image ci-dessus.

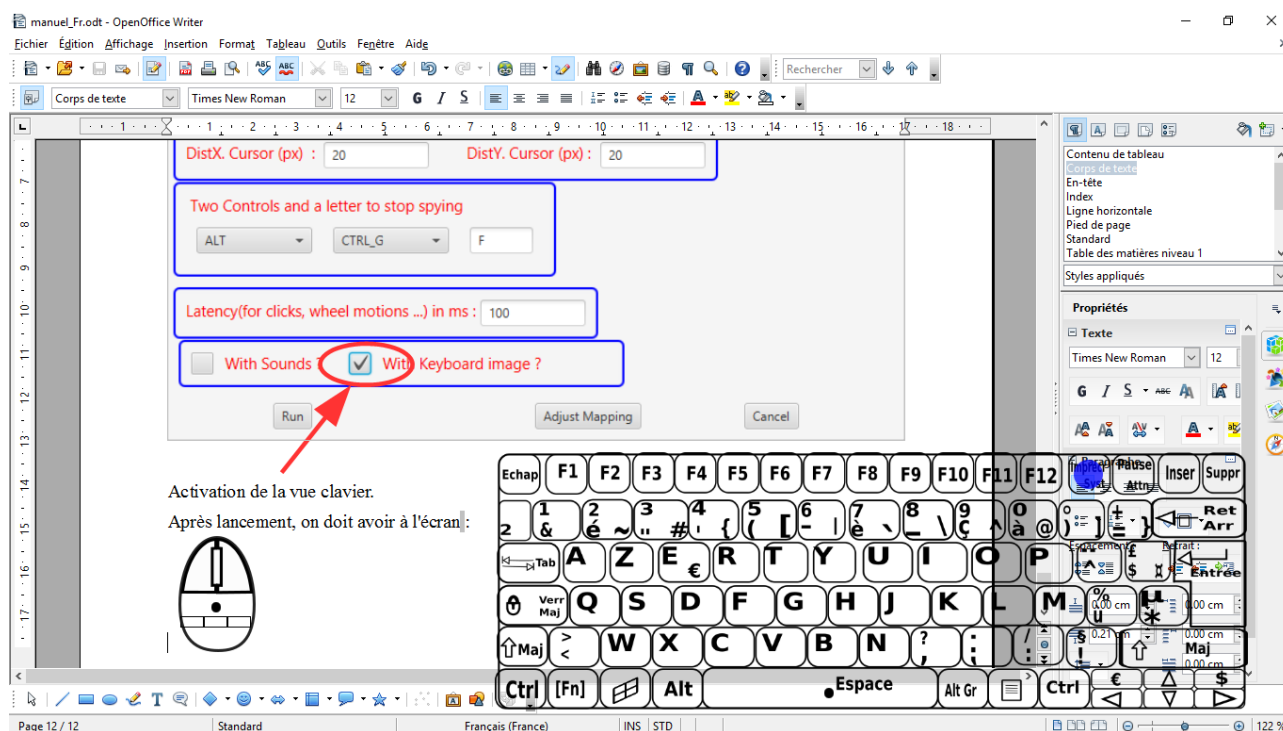
3.5 Activation de la visibilité du clavier

On va rester avec la souris de taille Small en mode mobile pour activer le clavier. Le comportement de la souris ne va pas différer par rapport aux 2 cas vus juste au dessus.

L'écran de configuration initial a donc l'aspect suivant :



Activation de la vue clavier après avoir choisi le bon clavier dans la liste de choix et la taille clavier. Après lancement, on doit avoir à l'écran :



En plus des actions souris, il apparaît sur l'image clavier des ronds rouges (Contrôle type CTRL,

ALT, ALTGR) et un rond bleu en fonction des touches pressées.

Le point noir au milieu de la barre espace permet de relocaliser l'image du clavier en cliquant une première fois dessus pour le faire devenir vert et puis en cliquant ailleurs dans l'écran pour déplacer cette image du clavier.

4 Annexe

4.1 Préambule

Si vous n'avez pas trouvé votre clavier, vous pourrez en rajouter un mais cette section va demander quelques connaissances informatiques, et notamment pour la création d'images de clavier, l'utilisation d'outil de dessins comme Inkscape : <https://inkscape.org/fr/telecharger/windows/>

il existe des versions pour Linux et OS X.

Pour la mise à l'échelle du clavier (800px * 275 px), on pourra utiliser PhotoFiltre 7 :

<http://www.photofiltre-studio.com/pf7.htm>

La configuration de la souris et des claviers est externalisée dans des fichiers textes localisés dans le répertoire **config**. Chaque ligne (qui n'est pas un commentaire) de ces fichiers est de type paire :

clé = valeur

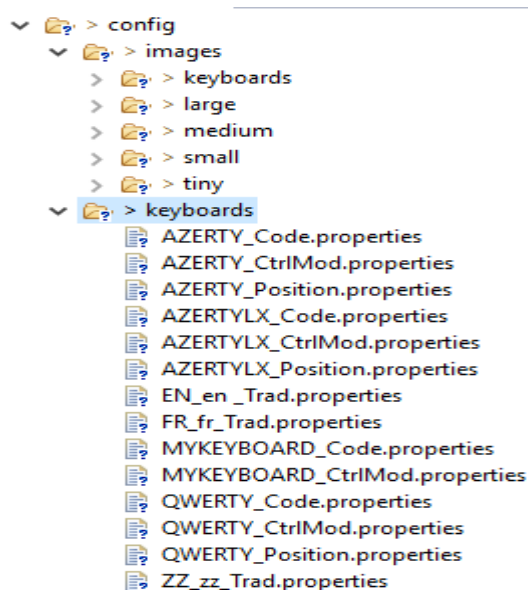
Pour ce qui concerne la souris, on ne gère que 3 boutons, le fichier de configuration est pour Windows :

config/windowsWheel.properties :

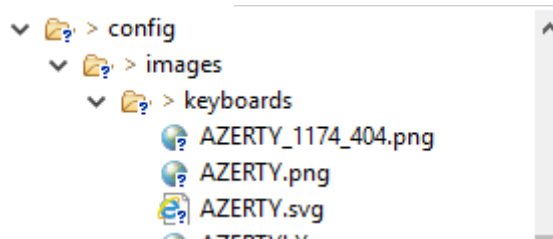
```
#Mapping Mouse L: Left/Gauche
# R : Right/Droit
# M: Middle/Milieu
#Tue Feb 28 08:57:31 CET 2017
3=M
2=R
1=L
```

On peut avoir, pour certaine souris, une inversion de rang entre le bouton central et le bouton droit. A corriger dans ce fichier.

config/keyboards :



config/images/keyboards :



Le fichier effectivement utilisé est, dans le cas du clavier AZERTY, **AZERTY.png**. Si vous n'utilisez que l'image de la souris dans vos démonstrations, vous pouvez vous passer de la création de l'image de votre clavier, mais il est nécessaire de configurer la partie mapping code/touche de votre clavier.

4.2 Le mapping du clavier

Répertoire : **config/keyboards**

Un point à noter est que les noms des fichiers de configuration sont très important, il faudra bien respecter la règle dans ce nommage en faisant attention à la casse (majuscule/minuscule).

4.2.1 Donner un identifiant au clavier

Comme dit plus, ce nom va constituer l'identifiant du clavier, et il doit bien sur être unique. Pour ce document, je vais prendre l'identifiant :

OCCITAN

Note : Pour information, il existe un logiciel qui à partir du clavier AZERTY rajoute des combinaisons de touches pour les lettres majuscules accentuées nécessaires à l'Occitan (kbdoc) :

<http://www.panoccitan.org/le-logiciel-de-clavier-pour-loccitan/>

le clavier est celui-ci :

z	1 &	2 é ~	3 é ~	4 { (5 [-	6 _	7 è \	8 ç ^	9 Ç ^	0 À °	1 + ±	
	A Á	Z Æ	E €	R ®	T	Y	U Ú	I Í	O Ó	P Ò	^ ¢	£ ¨
	Q	S	D	F	G	H	J	K	L	M	% ù	µ *
Maj	>	W «	X »	C ©	V	B	N Ñ	?	. :	/	\$!	Maj
										Alt Gr		

Le clavier kbdoc complet.

Nous n'allons pas réaliser réellement ce mapping, mais nous allons montrer comment on pourrait faire. Dans ce cas là , on pourrait partir du mapping AZERTY et le compléter des caractères manquant.

4.2.2 Les fichiers à créer pour le mapping clavier :

A partir donc du nom OCCITAN, pour ce qui concerne le clavier, il faudra créer :

- **config/keyboards/OCCITAN_Code.properties**
- **config/keyboards/OCCITAN_CtrlMod.properties**
- **config/keyboards/OC_oc_Trad.properties**

4.2.2.1 config/keyboards/OCCITAN_Code.properties

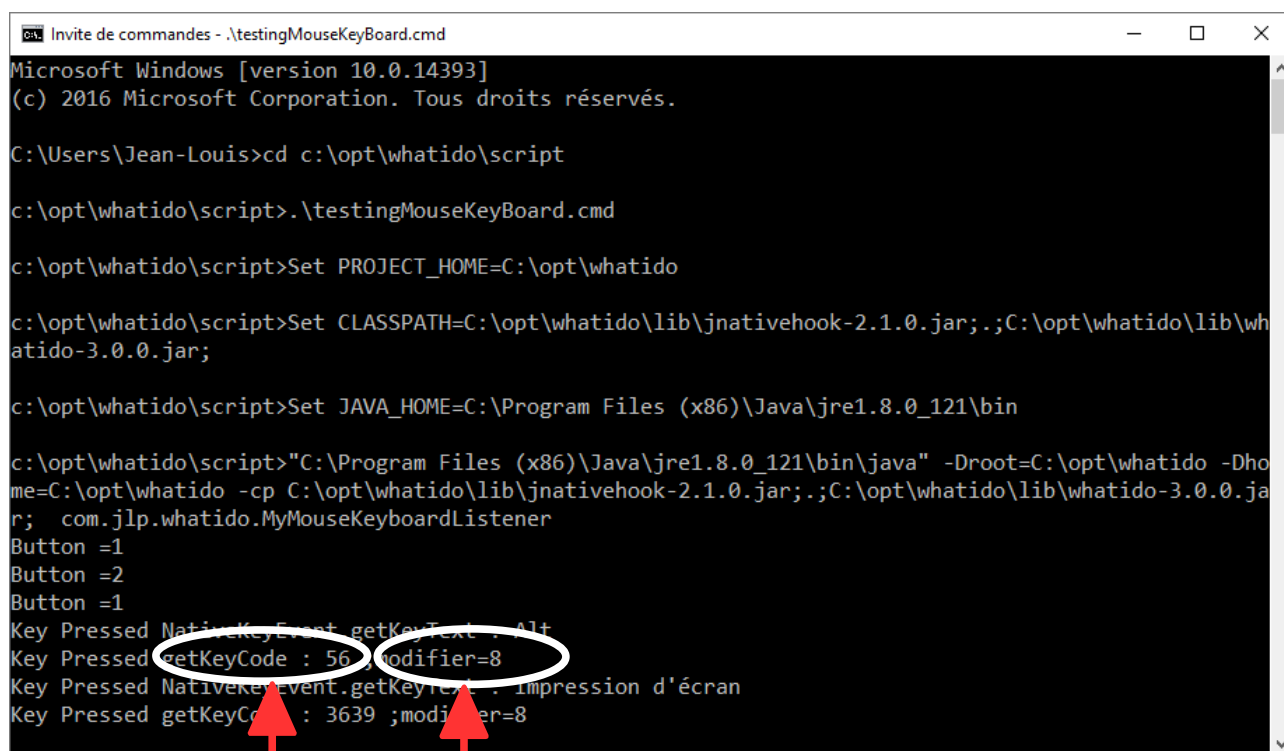
La structure de chaque ligne de ce fichier est donnée ci-dessous :

<code-touche>_<modifiers> = <identifiant-touche>

Les lignes commençant par # sont des commentaires, et ne sont pas prise en compte par whatido, lors de la lecture du fichier.

Comment obtient-on le **code-touche** et les **modifiers** ?

- Lancer une commande console windows,
- se déplacer sous **whatido/script**
- lancer **.\testingMouseKeyboard.cmd**
- puis fermer la fenêtre **helloWord**.



```

Microsoft Windows [version 10.0.14393]
(c) 2016 Microsoft Corporation. Tous droits réservés.

C:\Users\Jean-Louis>cd c:\opt\whatido\script

c:\opt\whatido\script>.\testingMouseKeyboard.cmd

c:\opt\whatido\script>Set PROJECT_HOME=C:\opt\whatido

c:\opt\whatido\script>Set CLASSPATH=C:\opt\whatido\lib\jnativehook-2.1.0.jar;.;C:\opt\whatido\lib\whatido-3.0.0.jar;

c:\opt\whatido\script>Set JAVA_HOME=C:\Program Files (x86)\Java\jre1.8.0_121\bin

c:\opt\whatido\script>"C:\Program Files (x86)\Java\jre1.8.0_121\bin\java" -Droot=C:\opt\whatido -Dhome=C:\opt\whatido -cp C:\opt\whatido\lib\jnativehook-2.1.0.jar;.;C:\opt\whatido\lib\whatido-3.0.0.jar; com.jlp.whatido.MyMouseKeyboardListener
Button =1
Button =2
Button =1
Key Pressed NativeKeyEvent.getKeyText : Alt
Key Pressed NativeKeyEvent.getKeyCode : 56 ;modifier=8
Key Pressed NativeKeyEvent.getKeyText : Impression d'écran
Key Pressed NativeKeyEvent.getKeyCode : 3639 ;modifier=8
  
```

On peut lire le code-touche et le modifier . Ne pas tenir compte de la ligne `getKeyText` (pb lié à une bibliothèque extérieure au produit)

Par exemple pour la touche a (a minuscule) :

```

Key Pressed NativeKeyEvent.getKeyText : A
Key Pressed NativeKeyEvent.getKeyCode : 30 ;modifier=0
  
```

On mettra dans le fichier

30_0 = a

Pour le A majuscule :

```
Key Pressed NativeKeyEvent.getKeyText : A  
Key Pressed getkeyCode : 30 ;modifier=1
```

soit

30_1 = A

Pour une autre touche comme Impression Ecran :

```
Key Pressed NativeKeyEvent.getKeyText : Impression d'écran  
Key Pressed getkeyCode : 3639 ;modifier=0
```

on mettra par exemple :

3639_0 = PRECR

La valeur PRECR est arbitraire et pourra ensuite être éventuellement traduite dans le fichier **OC_oc_Trad.properties** (voir plus bas) . On devra faire cela pour toutes les touches qui ne sont pas une lettre ou un chiffre (s'inspirer du fichier AZERTY_Code.properties pour le nommage de ces caractères particuliers)

Il faut donc passer toutes les touches du clavier sauf le pavé numérique, et pour chaque touche jouer avec les contrôles (CTRL, MAJ, ALTGR) pour avoir une ligne, dans le fichier de configuration, pour chaque caractère disponible sur le clavier. Pour un clavier AZERTY, on a à peu près 175 lignes (c'est un peu fastidieux) .

Faire bien le distinguo pour les touches en double (Maj Gauche et droit, Contrôle Gauche et Droit)

4.2.2.2 config/keyboards/OCCITAN_CtrlMod.properties

Ce fichier répertorie les contrôles et modifieurs du clavier. Il suffit de reporter ici la partie de code trouvée dans la configuration précédente pour ces seules touches avec l'inversion de la clé et de la valeur .

On devrait avoir un fichier identique à :

```
SHFT_L = 42_1  
CTRL_L = 29_2  
  
ALT = 56_8  
ALTGR = 56_130  
CTRL_R = 29_32  
SHFT_R = 3638_16
```

4.2.2.3 config/keyboards/OC_oc_Trad.properties

Ce fichier permet une traduction des caractères spéciaux pour un affichage dans l'un des 3 champs de la souris. Si on ne met pas de correspondance dans ce fichier, on affichera la valeur du fichier **OCCITAN_Code.properties**.

En java, il faut éviter de travailler directement avec des caractères propre à la langue (par exemple pour le Français, les caractères accentués, le c cédille ...). Ce fichier permet une traduction en un codage de caractère universel qui s'appelle unicode.

Exemple dans notre cas pour é (e accent aigu) :

On a noté dans notre fichier **OCCITAN_Code.properties** la ligne :

```
# é eacute
3_0 = EACUTE
```

Dans notre Fichier **OC_oc_Trad.properties** on aura :

```
# é eacute
EACUTE = \u00E9
```

Les codes unicodes sont constitués pour java de \u<codeHexaDuCaractere>

Le code Hexadécimal des caractères unicodes sont donnés là :

<https://unicode-table.com/fr/#control-character>

Un extrait : [=> \u005B

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0010	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	CS	RS	US
0020		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0060	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{		}	~	¸
0080	XXX	XXX	BPH	NBH	IND	NEL	SBA	EBA	HTB	HTJ	VTB	PLD	PLU	RI	SS2	SS3
0090	DC5	PU1	PU2	STB	OGH	MW	SPA	EPA	BOB	XXX	BCI	CSI	ST	DEC	PM	APC
00A0	NB SP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
00B0	°	±	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾		

4.3 La visualisation du clavier

Cette visualisation est réalisée à travers 2 fichiers :

- une image png du clavier pour clavier AZERTY :
config\images\keyboards\AZERTY.png taille (800*275 px)
- un fichier de configuration pour la localisation des touches sur l'image pour clavier AZERTY : **config\keyboards\AZERTY_Position.properties**

Dans notre cas, il faudra remplacer le nom AZERTY par OCCITAN.

4.3.1 Image png du clavier OCCITAN

Le but est d'obtenir une image **config\images\keyboards\OCCITAN.png** d'une taille de 800*275 px (cette taille n'est pas obligatoire, mais elle est correcte pour les écrans ou vidéo-projecteurs)

L'explication de l'utilisation de Inskape et PhotoFiltre est hors du champ de ce document.

On peut partir du fichier **config\images\keyboards\AZERTY.svg** que l'on copie en **config\images\keyboards\OCCITAN.svg**.

On ouvre ensuite avec Inskape, et on déplace/rajoute/supprime les caractères des touches pour les remettre sur la bonne touche et à sa place (haut/bas/droite/gauche) dans la touche pour ressembler au clavier réel.

Ensuite on exporte la partie du clavier au format png sous le même répertoire en donnant la taille dans le nom : ex **config\images\keyboards\OCCITAN_1100*400.png**

On ferme Inskape en sauvegardant au format **svg**, on lance PhotoFiltre et on charge l'image **config\images\keyboards\OCCITAN_1100*400.png**

On duplique l'image, puis on passe la taille de l'image à 800*275 px (On peut activer la distorsion pour avoir cette taille).

On enregistre ensuite sous **config\images\keyboards\OCCITAN.png**.

Vous devez ensuite dupliquer image 800*275 avec une taille de 400*138 et l'appeler **OCCITAN_S.png**

Vous devez ensuite dupliquer image 800*275 avec une taille de 600*206 et l'appeler **OCCITAN_M.png**

4.3.2 Localisation des touches dans l'image

Le fichier **config\keyboards\OCCITAN_Position.properties** servira au produit à repérer les coordonnées des touches frappées dans l'image du clavier. On le réalise avec l'image normale 800px * 275px

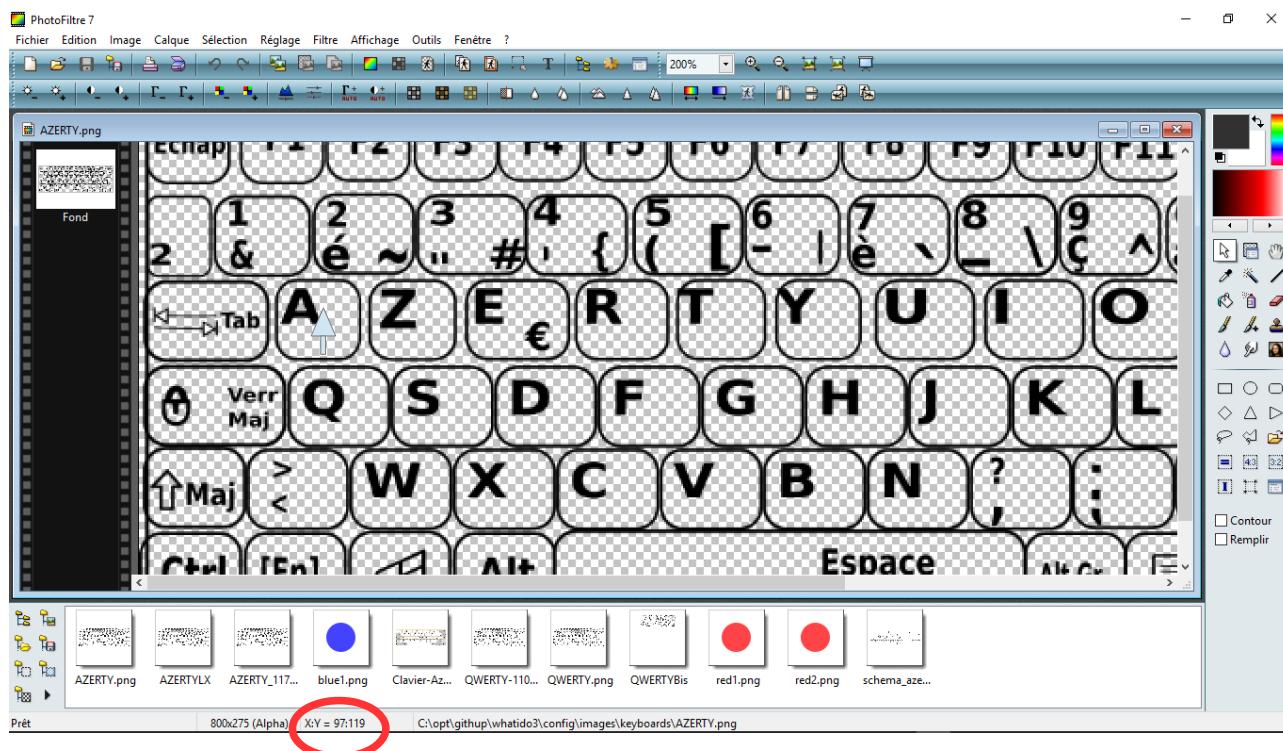
La structure de chaque ligne de ce fichier est la suivante :

CodeTouche_Modifier = X_Y sur image clavier (800px*275px) Fichier OCCITAN.png

Pour remplir ce fichier, il faut :

- lancer l'utilitaire **.testingMouseKeyboard.cmd** déjà vu plus haut
- charger l'image OCCITAN.png dans PhotoFiltre.
- Puis se positionner sur chaque touche et noter les coordonnées fournies par PhotoFiltre. Voir

image ci-dessous.



Par exemple en positionnant le curseur sur la touche A (bien se positionner au centre de la touche) , on peut lire les coordonnées suivantes :

X:Y = 97:119

On aura alors dans le fichier **config\keyboards\OCCITAN_Position.properties** la ligne suivante :

```
# A
30_0 = 97_119
```

Il faut ainsi paramétrer toutes les touches du clavier (hors modificateurs/modifiers) sans tenir compte des possibilités liés aux modificateurs (on aura pour les caracteres le modificateur par défaut 0) Cependant pour les touches de contrôles, il faudra rajouter les possibilités de double touches (ALT + CTRL_G ...) en incluant le modificateur , voir le fichier **AZERTY_Position.properties**.

Exemple pour un modificateur Controle Droit :

```
# Ctrl D Ctrl R
29_32 = 607_250
29_40 = 607_250
29_33 = 607_250
```

4.4 Mettre du Son

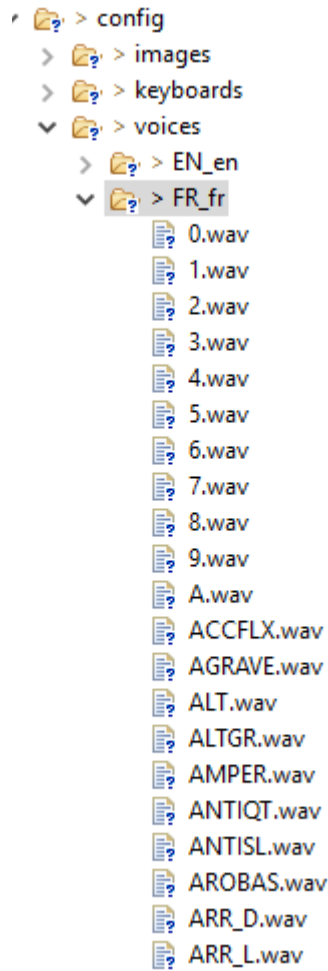
Pour chaque action faite sur la souris ou le clavier, on peut rajouter une description sonore. Les fichiers sonore sont au format .wav.

Pour enregistrer les sons, on peut le faire de deux façons :

- enregistrer à l'aide du micro de votre ordinateur et un logiciel d'enregistrement ([Audacity](#))
- utiliser un logiciel de type Text To Speech, c'est ce que j'ai fait avec [balabolka](#)

4.4.1 Fichiers de configurations

L'arborescence de la configuration des voix est la suivante :



Les fichiers **.wav** doivent se trouver sous le répertoire :
config/voices/OC_oc pour notre exemple.

Le préfixe du nom de fichier correspond à la clé de la ligne correspondante du fichier :

config/keyboards/OCCITAN_Code.properties