

MyScreenRecorder

Pour capturer votre écran d'ordinateur et l'audio.

Table des matières

1 Contexte.....	3
2 Architecture logicielle.....	3
2.2 ffmpeg.....	3
2.3 openjdk11.....	4
2.4 openJFX11.....	4
2.5 myscreenrecorder.....	4
3 Procédure.....	5
3.2 Récupération archive de Github.....	5
3.3 Création du répertoire application et extraction des répertoires.....	5
3.4 Création du répertoire de création des vidéos :.....	6
3.5 Configuration des interfaces audios.....	6
3.5.1 Cas du micro et hauts parleurs intégrés.....	6
3.5.2 Cas du micro et hauts parleurs intégrés.....	7
4 Enregistrement plein écran.....	9
5 Enregistrement zone rectangulaire.....	10

Introduction

1 Contexte

Sous Windows, je n'ai pas pu (ou su) trouvé de logiciel gratuit simple pour faire de la capture d'écran et audio sous forme de vidéo. Très utile pour faire des tutoriels par exemple.

Il en existe de gratuit mais la durée d'enregistrement est limité ou bien il y a insertion d'un filigrane en début et fin d'enregistrement. D'autres génèrent des fichiers avi non compressés (très gros) avec une limite de taille à 2 Go.

Sous Linux, j'en connais au moins 2 :

- [RecordMyDesktop](http://recordmydesktop.sourceforge.net/downloads.php) => <http://recordmydesktop.sourceforge.net/downloads.php>
- [SimpleScreenRecorder](https://www.maartenbaert.be/simplescreenrecorder/) => <https://www.maartenbaert.be/simplescreenrecorder/>

Bien que mon logiciel puisse fonctionner sous Linux, je vais me consacrer plutôt à son utilisation sous Windows. Je fournis tous les modes opératoires et scripts pour le faire fonctionner aussi sous Linux, sans détailler.

2 Architecture logicielle

Important : je vais décrire une installation de **myscreenrecorder** sous le répertoire au choix :

C:\opt\myscreenrecorder

D:\opt\myscreenrecorder

Si on respecte les appellations décrites ci-dessous, on n'aura pas à toucher aux scripts de lancement.

2.2 ffmpeg

C'est le cœur de l'outil **myscreenrecorder**, le framework **ffmpeg** de traitement de vidéo est aussi capable de faire de la capture d'écran et d'audio, mais les commandes en lignes peuvent devenir très complexe.

Pour la version Windows, il faut prendre la version binaire static (cad incluant toute les bibliothèques) => Le site de téléchargement est [là](https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-latest-win64-static.zip) =>

<https://ffmpeg.zeranoe.com/builds/win64/static/ffmpeg-latest-win64-static.zip>

A noter qu'avec Linux, il vaut mieux utiliser le package ffmpeg propre à votre distribution. J'ai eu des problèmes avec la version static.

A dézipper sous \opt (sous C : ou D : selon le cas) ; le dossier doit s'appeler **\opt\ffmpeg-latest-win64-static**

2.3 openjdk11

J'utilise openjdk 11.0.2 pour cette version de **myscreenrecorder** à télécharger [ici](https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2_windows-x64_bin.zip) => https://download.java.net/java/GA/jdk11/9/GPL/openjdk-11.0.2_windows-x64_bin.zip

A dézipper sous \opt (sous C : ou D : selon le cas) ; le dossier doit s'appeler **\opt\jdk-11.0.2**

2.4 openJFX11

J'utilise openJFX-SDK 11.0.2 pour cette version de **myscreenrecorder** à télécharger [ici](http://gluonhq.com/download/javafx-11-0-2-sdk-windows/) => <http://gluonhq.com/download/javafx-11-0-2-sdk-windows/>

A dézipper sous \opt (sous C : ou D : selon le cas) ; le dossier doit s'appeler **\opt\javafx-sdk-11.0.2**

2.5 myscreenrecorder

Il s'agit de l'application proprement dite qui est en fait un front-end graphique de **ffmpeg** pour l'enregistrement de l'écran et éventuellement de l'audio à travers un micro ou à travers les haut-parleurs utilisés (casque ou propre au pc).

En plus des logiciels déjà cités, j'utilise aussi une bibliothèque (fournie avec le package github) d'interface JNI **JnativeHook** disponible [ici](https://github.com/kwhat/jnativehook/releases/download/2.1.0/jnativehook-2.1.0.zip) =>

<https://github.com/kwhat/jnativehook/releases/download/2.1.0/jnativehook-2.1.0.zip>

INSTALLATION

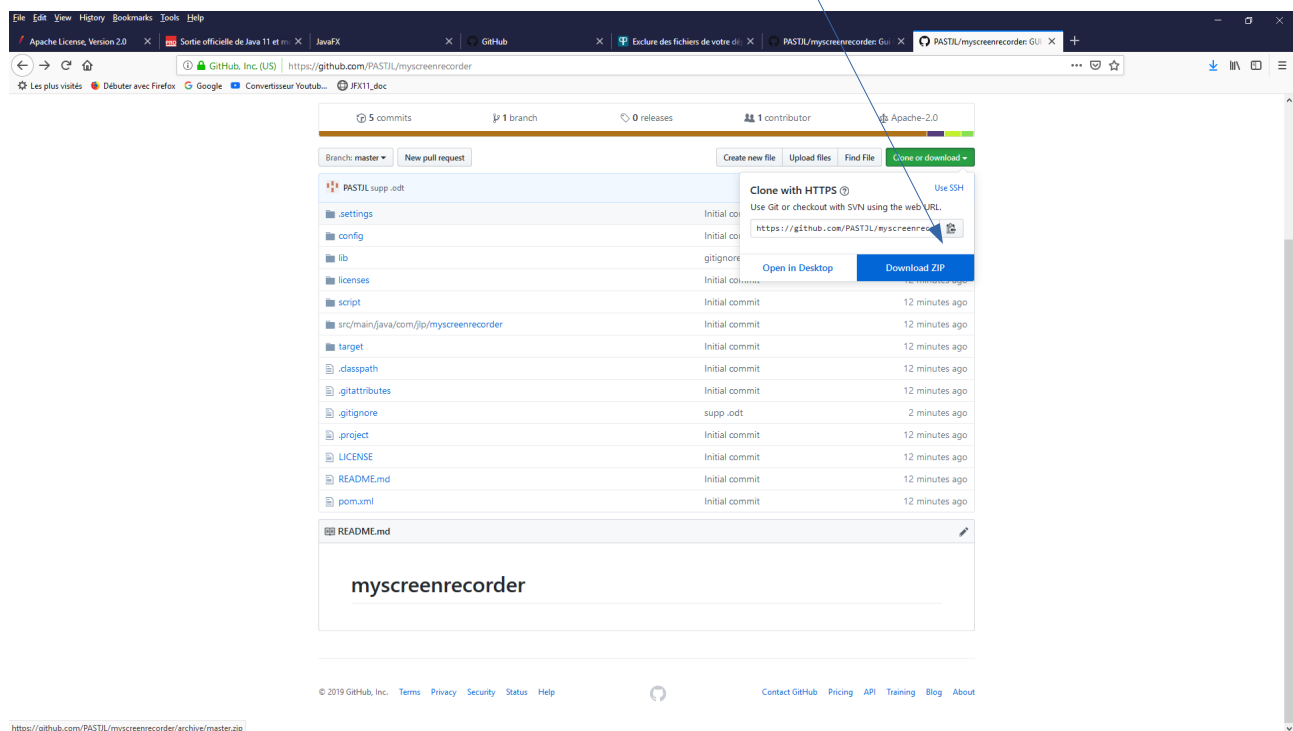
3 Procédure

On suppose que **ffmpeg**, **openjdk** et **openjfx** ont été installés conformément au paragraphe 2 ci-dessus sous le répertoire (**C:** ou **D:**) \opt\

3.2 Récupération archive de Github

Aller sur le site [Github](https://github.com) pour récupérer l'archive zip du projet :

<https://github.com/PASTJL/myscreenrecorder>



3.3 Création du répertoire application et extraction des répertoires

Créer un répertoire (**C:** ou **D:**)\opt\myscreenrecorder

Extraire du fichier zip téléchargé (**myscreenrecorder-master.zip**) les répertoires :

- **lib**
- **script**

- **config**
- **licences**
- **manuals**

dans le répertoire (C: ou D:)\opt\myscreenrecorder

Créer un raccourci sur le bureau du fichier :

(C: ou D:)\opt\myscreenrecorder\script\myscreenrecorder.cmd

et y associer l'icône (C: ou D:)\opt\myscreenrecorder\script\myscreenrecorder.ico

3.4 Création du répertoire de création des vidéos :

Créer un répertoire (C: ou D:)\opt\screenRecordVideos

3.5 Configuration des interfaces audios

C'est la partie la plus délicate de l'installation.

Dans mon cas en plus du micro et haut-parleurs intégrés au pc, j'utilise un casque avec micro.

Il y a donc plusieurs interfaces possibles, mais une seule doit rester active lors de l'enregistrement.

Pour prévenir toute erreur, faire une copie de sauvegarde du fichier \opt\myscreenrecorder\config\config.properties

Avec un éditeur de texte (notepad, wordpad ...) ouvrir le fichier \opt\myscreenrecorder\config\config.properties

```

4 # .\ffmpeg -list_devices true -f dshow -i dummy
5 # choose the alternative name to avoid accent problems and Don't forget to escape anti slash if present in the name of the device
6 #USB 2
7 #win_devmicroCasque="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{C3C6F4F9-FAE9-41BF-A18A-6A90BB262BEA}"
8
9 #USB1
10 # Debut des lignes à modifier
11 win_devmicroCasque="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
12
13 win_devspeakerpc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
14 win_devmicropc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"
15 # You must determine what is the alsa audio header input for me hw:2,0 and the same for the native speakers
16 lin_devmicroCasque="hw:2,0"
17 lin_devspeakerpc="TODO"
18 win_list_audio=None;microCasque;speakerpc;micropc;
19 lin_list_audio=None;microCasque;
20
21 #Fin des lignes à modifier
22

```

Les lignes à modifier sont surlignées en jaune, on ne s'intéresse qu'à la partie Windows.

3.5.1 Cas du micro et haut-parleurs intégrés

Dans ce cas là, il faudra trouver les valeurs win_devspeakerpc et win_devmicropc .

Pour cela, aller dans le répertoire \opt\myscreenrecorder\script

Lancer .\searchAudioDevices.cmd on obtient ce type de sortie :

```

Invite de commandes

D:\opt\myscreenrecorder\script>D:\opt\ffmpeg-latest-win64-static\bin\ffmpeg -list_devices true -f dshow -i dummy
ffmpeg version N-93580-g036b4b0f85 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 8.3.1 (GCC) 20190414
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --
enable-libluray --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus
--enable-libshine --enable-libsnappy --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --
enable-libx264 --enable-libx265 --enable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --e
nable-libvo-amrwbenc --enable-libmysofa --enable-lspspeex --enable-libxvid --enable-libaom --enable-libmfx --enable-amf --enable-ffnvcodec --enable-c
uvid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt
                                libavutil      56. 26.100 / 56. 26.100
                                libavcodec     58. 51.100 / 58. 51.100
                                libavformat   58. 27.102 / 58. 27.102
                                libavdevice   58.  7.100 / 58.  7.100
                                libavfilter    7. 48.100 /  7. 48.100
                                libswscale     5.  4.100 /  5.  4.100
                                libswresample  3.  4.100 /  3.  4.100
                                libpostproc   55.  4.100 / 55.  4.100
[dshow @ 0000029da7388f00] DirectShow video devices (some may be both video and audio devices)
[dshow @ 0000029da7388f00] "HD WebCam"
[dshow @ 0000029da7388f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
Global!
[dshow @ 0000029da7388f00] DirectShow audio devices
[dshow @ 0000029da7388f00] "Mixage st[er]eo (Realtek High Definition Audio)"
[dshow @ 0000029da7388f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"
[dshow @ 0000029da7388f00] "R[ec]seau de microphones (Realtek High Definition Audio)"
[dshow @ 0000029da7388f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"
dummy: Immediate exit requested
D:\opt\myscreenrecorder\script>

```

Remarque pour Linux, la commande à lancer est arecord -l, et l'interface est du type hw:1,0 (1 et 0 à adapter en fonction du numéro de carte et de device voir <https://trac.ffmpeg.org/wiki/Capture/ALSA>)

Ce qui va nous intéresser, ce sont les interface audios entourés en rouge :

Dans notre cas on devra remplir la valeur de l'**alternative name** pour éviter les problèmes de caractères accentués.

Haut-parleurs :

win_devspeakerpc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B0B786F-15F0-4233-818D-FFC7077ACDED}"

Important, dans notre cas, il faudra doubler l'anti-slash avant wave (et éventuellement ailleurs s'il est présent) sinon l'interface n'est pas reconnue.

Micro PC :

win_devmicropc="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"

Important, dans notre cas, il faudra doubler l'anti-slash avant wave (et éventuellement ailleurs s'il est présent) sinon l'interface n'est pas reconnue.

Et puis la liste des devices audios doit être :

win_list_audio=None;speakerpc;micropc;

Enregistrer le fichier modifié **\opt\myscreenrecorder\config\config.properties**

3.5.2 Cas du micro casque

Si on utilise un casque micro USB, on veillera à toujours prendre la même sortie USB.

On branche notre casque et l'on procède comme indiqué au paragraphe 3.5.1.

La sortie de la commande : **.\searchAudioDevices.cmd**

est alors :

```

Invite de commandes
ffmpeg version N-93580-g036b4b0f85 Copyright (c) 2000-2019 the FFmpeg developers
  built with gcc 8.3.1 (GCC) 20190414
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-fontconfig --enable-gnutls --enable-iconv --enable-libass --enable-libdav1d --
  enable-libbluay --enable-libfreetype --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus
  --enable-libshine --enable-libsnap --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack --enable-libwebp --
  enable-libx264 --enable-libx265 --enable-libxml2 --enable-libzimg --enable-lzma --enable-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --e
  nable-libvo-amrwbenc --enable-libmysofa --enable-libspeex --enable-libxvid --enable-libaom --enable-libmfx --enable-amf --enable-ffnvcodec --enable-c
  uid --enable-d3d11va --enable-nvenc --enable-nvdec --enable-dxva2 --enable-avisynth --enable-libopenmpt
  libavutil 56. 26.100 / 56. 26.100
  libavcodec 58. 51.100 / 58. 51.100
  libavformat 58. 27.102 / 58. 27.102
  libavdevice 58. 7.100 / 58. 7.100
  libavfilter 7. 48.100 / 7. 48.100
  libswscale 5. 4.100 / 5. 4.100
  libswresample 3. 4.100 / 3. 4.100
  libpostproc 55. 4.100 / 55. 4.100
[dshow @ 00002343eea8f00] DirectShow video devices (some may be both video and audio devices)
[dshow @ 00002343eea8f00] "HD WebCam"
[dshow @ 00002343eea8f00] Alternative name "@device_pnp_\\?\\usb#vid_0bda&pid_5621&mi_00#6&199da46c&0&0000#{65e8773d-8f56-11d0-a3b9-00a0c9223196}
\\global"
[dshow @ 00002343eea8f00] DirectShow audio devices
[dshow @ 00002343eea8f00] "Microphone (Plantronics .Audio 400 DSP)"
[dshow @ 00002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
[dshow @ 00002343eea8f00] "Realtek High Definition Audio"
[dshow @ 00002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{4B08786F-15F0-4233-818D-FFC7077ACDED}"
[dshow @ 00002343eea8f00] "Réseau de microphones (Realtek High Definition Audio)"
[dshow @ 00002343eea8f00] Alternative name "@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{A53D5F12-6864-4A6C-B6E1-D21EC139A79B}"
dummy: Immediate exit requested
D:\opt\myscreenrecorder\script>

```

Pour mon casque Plantronics, le micro est identifié et il faut paramétrer :

```
win_devmicrocasque="@device_cm_{33D9A762-90C8-11D0-BD43-00A0C911CE86}\\wave_{0764C06E-1C55-44E7-B6F1-0D138DEA352A}"
```

Important, dans notre cas, il faudra doubler l'anti-slash avant wave (et éventuellement ailleurs s'il est présent) sinon l'interface n'est pas reconnue.

Et puis la liste des devices audios doit être :

```
win_list_audio=None;microcasque;speakerpc;micropc;
```

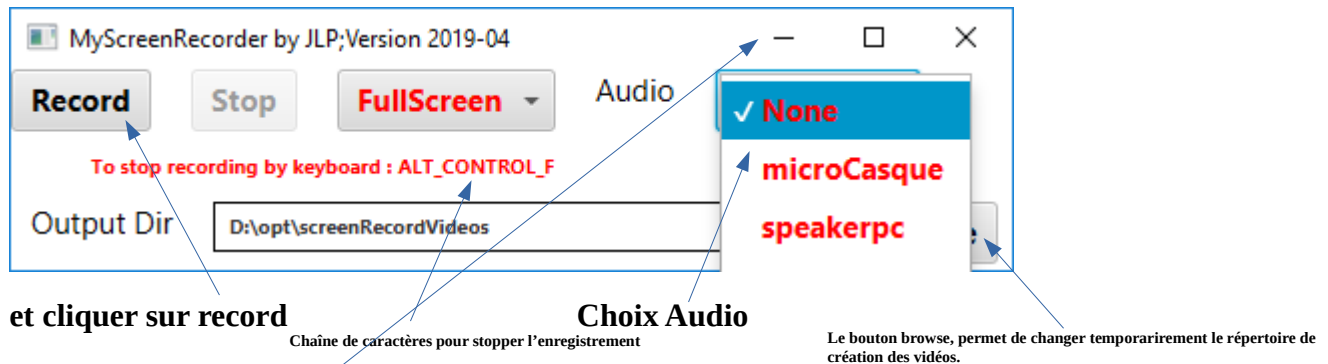
Enregistrer le fichier modifié `\opt\myscreenrecorder\config\config.properties`

Et voilà **myscreenrecorder** est prête à fonctionner.

UTILISATION

4 Enregistrement plein écran

Cliquer sur l'icône de bureau myscreenrecorder puis choisir votre sortie audio ou pas



puis réduire la fenêtre.

Pour arrêter, soit taper la suite de caractère de fin d'enregistrement , ou ré-appeler la fenêtre dans la barre des taches et cliquer sur stop.

Enregistrement terminé, fichier resultat => D:\opt\screenRecordVideos\capture-2019-04-16-19-13-24.mkv2019-04-16-19-15-52.mkv
Cliquez pour fermer

Le popup vous indique l'emplacement du fichier résultats que l'on peut voir dans notre logiciel vidéo préféré (Windows Media Player, VLC).

Si l'on veut changer le répertoire de sortie vidéo de manière définitive, il faut modifier la variable **outputDir** dans le script `\opt\myscreenrecorder\script\myscreenrecorder.cmd`

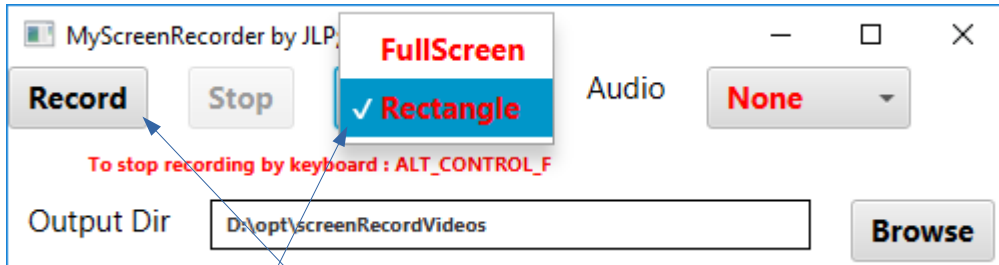
Changer la partie surlignée en jaune.

```
1 if exist D:\opt\ (set Disk=D) ELSE (set Disk=C)
2 Set PROJECT_HOME=%Disk%:\opt\myscreenrecorder
3 Set outputDir=%Disk%:\opt\screenRecordVideos
4 Set JAVA_HOME=%Disk%:\opt\jdk-11.0.2
5 Set JFX_HOME=%Disk%:\opt\javafx-sdk-11.0.2
6 Set pathffmpeg=%Disk%:\opt\ffmpeg-latest-win64-static
7 Set CLASSPATH=%PROJECT_HOME%\lib\jnativehook-2.1.0.jar,
```

On peut ensuite retravailler la vidéo dans un logiciel d'édition vidéo comme OpenShot, ShotCut ou Kdenlive ou autres logiciels commerciaux.

5 Enregistrement zone rectangulaire

Cela demande une opération supplémentaire de dessin du rectangle à enregistrer par drag and drop .



On choisit le rectangle , et puis comme ci dessus on peut choisir l'interface audio.

Et ensuite on clique sur record.

L'enregistrement ne démarre pas, il faut définir le rectangle en partant d'en haut à gauche (clic gauche sans relacher) vers le bas à droite (drag sans relacher) , puis relâcher. L'enregistrement démarre sur le rectangle dessiné.



Enfin, il faut définir le rectangle en partant d'en haut à gauche (clic gauche sans relacher) vers le bas à droite (drag sans relacher) , puis relâcher. L'enregistrement démarre sur le rectangle dessiné.

En fin d'enregistrement, il est indiqué aussi l'emplacement du fichier résultat.