animation, imagemagick, gifsicle, ffmpeg, gif, em

# La fabrique de GIF

Méthodes amusantes pour fabriquer des gif animés amusants

## Logiciels de dessin

Certains logiciels de dessin et de retouche d'image proposent des fonctions d'animation, c'est le cas pour **Gimp** et **Krita**

### Gimp

Des tutos pour y arriver avec GIMP :

- à lire : https://wiki.labomedia.org/index.php/Gif_anim%C3%A9.html
- en vidéo
    - Gimp facile : GIMP 2.10 : gif animé - https://www.youtube.com/watch?v=k0gr_XcT1EA
    - Night Evans : Comment faire un gif animé avec gimp - https://www.youtube.com/watch?v=Tnjs_dcd1lY

### Krita

(à compléter)

- https://docs.krita.org/en/user_manual/animation.html
- https://docs.krita.org/fr/reference_manual/render_animation.html

## Logiciels d'animation

(à compléter) Il existe des logiciels libres dédiés à l'animation

- comparatif : https://fosspost.org/alternative-software/end-users/open-source-2d-animation-software
- https://www.synfig.org/
- https://opentoonz.github.io/e/index.html
- https://morevnaproject.org/
- https://sourceforge.net/projects/tupi2d/
- https://www.pencil2d.org/

## Copies d'écran vidéo / Screencast

https://en.wikipedia.org/wiki/Comparison_of_screencasting_software

### Un petit bout de l'écran en vidéo avec Peek

Peek sur linux, est facile à utiliser pour transformer un petit bout d'écran en gif (mais aussi vers d'autres formats de fichiers comme: webm, mp4, apng) et fonctionne correctement avec un navigateur sur les plateformes de vidéo (testé au moins avec youtube). Avec peek on peut définir les dimensions de la capture, le nombre d'images par seconde

- installation, etc. : https://github.com/phw/peek
- https://en.wikipedia.org/wiki/Peek_(software)

## Petites manipulations

Des petits utilitaires en ligne de commande peuvent rendre de grands services

### Obtenir des infos sur un gif

Plusieurs possibilités pour des infos différentes

```
gifsicle -I anim.gif             # dimensions, nombe de frames, délai entre chaque frame (peut être différent), etc.
exiftool anim.gif                # dimensions, nombre de frames, durée totale, etc.
exiftool -b -FrameCount anim.gif # on peut avoir tous les champs séparément, ici sans label
identify anim.gif                # (identify fait partie d'imagemagick) dimensions, nombre de frames, etc.
ffprobe anim_out.gif             # dimensions, fps, peu complet
```

## Gifsicle

### Quelque exemples

### Enlever une image sur deux

La vitesse de lecture sera doublée, il faut d'abord connaître le nombe d'images. (source)

```
# 20200504 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle -I anim_in.gif # chercher le nombre d'images : 54
gifsicle -U anim_in.gif `seq -f "#%g" 0 2 54` -O2 -o anim_out.gif  # remplacer 54 par le nombre d'images
```

En détail :
**-U** (–unoptimize) : rendre indépendante chaque frame de l'animation d'origine. (Dans un gif optimisé, seuls les pixels qui changent sur l'image en cours sont mis à jour)
**-O2** : ré-optimiser l'animation de destination, une fois les images enlevées
La partie entre backticks (`) permet de créer une liste de numéros de frames à conserver, de 0 jusqu'à 54 en prenant un nombre sur deux, la partie `-f "#%g"` formate ces nombres correctement pour que gifsicle les interprête corectement comme des numéros de frames

### Enlever une image sur quatre sans changer la vitesse

Il faut connaître le nombre d'images et le délai entre chaque

```
# 20200504 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle -I anim_in.gif # chercher le nombre d'images : 54 et la vitesse 0.13
gifsicle --delay 52 -U eros_thanatos_kalimotxo_1.gif `seq -f "#%g" 0 4 54` -O2 -o anim_out.gif
```

**–delay 52** : indique la durée entre chaque frame en 1/100e de secondes.
La partie entre backticks (`) permet de créer une liste de numéros de frames à conserver, de 0 jusqu'à 54 en prenant un nombre sur 4.

### Supprimer des images

```
# 20200504 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle -U anim_in.gif --delete "#1" "#2" "#3" "#5" "#6" "#7" "#8" > anim_out.gif
```

### Supprimer une série d'images

```
# 20200504 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle anim_in.gif --delete '#70-74'> anim_out.gif # supprimer les images 70 à 74 (incluses)
```

### Réduire le nombre de couleurs

```
# 20200504 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle anim_in.gif -O3 --colors 32 > anim_out.gif
```

### Réduire la définition

```
# 20201109 / gifsicle v1.88 / Debian Stretch 9.5 @ Kirin
gifsicle --resize 300x300 -i anim_in.gif > anim_out.gif
# ou
gifsicle --scale 0.5 -i anim_in.gif > anim_out.gif
```

### Ressources gifsicle

- site du projet : https://www.lcdf.org/gifsicle/
- documentation en anglais : https://www.lcdf.org/gifsicle/man.html
- documentation en français (version de 2011) : http://www.traduction.cc/traduction/Manuel-Gifsicle-12.html

## Imagemagick

Imagemagick permet de réaliser et d'optimiser une animation gif.

### Quelques exemples

#### Faire une feuille de sprites (*spritesheet*) à partir d'un gif

Pratique pour se rendre compte des qualités de l'animation

```
# 20200505 / ImageMagick 6.9.7-4 Q16 x86_64 20170114 / Debian Stretch 9.5 @ Kirin
montage -coalesce anim.gif -tile 5x2 -geometry +0+0 -alpha On -background "rgba(0, 0, 0, 0.0)" -quality 100 anim_sprites.png
```

Ce qui produira une feuille de 5 images de large sur 2 images de haut, sur un fond noir

Un sujet de discussion avec des exemples pour manipuler des spritesheets / spritestrips, dans les deux sens.

#### Transformer une feuille de sprites en animation

```
# 20200505 / ImageMagick 6.9.7-4 Q16 x86_64 20170114 / Debian Stretch 9.5 @ Kirin
convert sprites.png -crop 165x165 +adjoin +repage -adjoin -loop 0 -set delay 200 sprites_out.gif
```

Produira une animation en boucle infinie où chaque image durera 2 secondes (**-set delay 200** : durée en 1/100e de secondes)

#### Transformer un gif en une séquence d'images

```
# 20200505 / ImageMagick 6.9.7-4 Q16 x86_64 20170114 / Debian Stretch 9.5 @ Kirin
convert -coalesce anim.gif anim_image_%03d.png
```

#### Transformer une séquence d'images en gif

Les images doivent avoir des noms de fichiers adaptés : image_000.png, image_001.png, image_002.png. Ce n'est pas nécessaire que les nombres se suivent, il faut juste respecter un ordre croissant.

```
# 20200510 / ImageMagick 6.9.7-4 Q16 x86_64 20170114 / Debian Stretch 9.5 @ Kirin
convert -delay 8 -loop 0 image_*.png anim.gif
```

-delay 8 : un délai de 8/100e de secondes entre chaque image, soit environ 12 images / seconde
-loop 0 : boucler indéfiniment

Si l'animation doit avoir une taille différente des images de départ, on peut la redimensionner :

```
# 20200510 / ImageMagick 6.9.7-4 Q16 x86_64 20170114 / Debian Stretch 9.5 @ Kirin
convert -delay 8 -loop 0 image_*.png -scale 100x100 anim.gif
```

D'autres opérateurs peuvent être utiles selon le type des images d'origine : pour réduire le nombre de couleurs (-colors), optimiser les couches (-optimize), ajouter du flou (-fuzz), etc.

### Ressources Imagemagick

- Bases des animations
- Modification des animations
- Optimisation des animations

## ffmpeg

#### Transformer une vidéo en animation gif source

```
# 20200504 / gifsicle v1.88, ffmpeg version 3.2.14-1~deb9u1 / Debian Stretch 9.5 @ Kirin
ffprobe video.mp4 # infos sur la vidéo : codecs, dimensions, framerate, etc.
ffmpeg -i video.mp4 -s 160x90 -pix_fmt rgb24 -r 2 -f gif - | gifsicle --optimize=3 --delay=4 > videoanim.gif
```

Dans un premier temps un fichier vidéo est redimensionné (**-s 160×90**), on en extrait 2 images par secondes (**-r 2**) pour créer un gif, ce gif est envoyé à gifsicle pour être optimisé.

#### Transformer une vidéo en séquence d'images

# 20200504 / gifsicle v1.88, ffmpeg version 3.2.14-1~deb9u1 / Debian Stretch 9.5 @ Kirin

```
ffmpeg -i robot.mp4 -r 1 -f image2 "image-robot-%4d.png"
```

#### Transformer une vidéo en planche d'iamge (***imagesheet***) source

```
# 20200504 / ffmpeg version 3.2.14-1~deb9u1 / Debian Stretch 9.5 @ Kirin
ffprobe video.mp4 # infos sur la vidéo : fps, durée
ffmpeg -i video.mp4 -frames 1 -vf "select=not(mod(n\,30)),scale=160:90,tile=10x18" planche.png
```

La vidéo fait 180s à 29.97 fps, si on veut une image par seconde, il y aura donc 180 images (**tile=10×18**) et on prend une image toutes les 30 images (nombre entier nécessaire) avec **mod(n\,30)**

Il existe des scripts tous faits pour réaliser ce genre de choses, comme vcs :

- http://p.outlyer.net/vcs
- https://unix.stackexchange.com/questions/63769/fast-tool-to-generate-thumbnail-video-galleries-for-command-line
- https://stackoverflow.com/questions/31223926/ffmpeg-command-to-create-thumbnail-sprites
- https://superuser.com/questions/538112/meaningful-thumbnails-for-a-video-using-ffmpeg

### Ressources

- détails sur le format de fichier GIF : http://giflib.sourceforge.net/whatsinagif/bits_and_bytes.html

## Divers services en ligne

**Piskel** https://www.piskelapp.com/

Editeur de sprites utilisable dans le navigateur, permet d'ouvrir un gif animé et de le transformer pixel par pixel!

(à compléter)

- https://ezgif.com/
- https://charlesstover.com/spritesheet2gif/
- https://is.si/animator/ (Sprite Sheet Animator)
- https://jacklehamster.github.io/utils/gif2sprite/

## Exemples

Cube en rotation façon tricot
Transformer un texte en animation

## GIFever en temps de crise

Transportés ici GIFever en temps de crise

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/logiciel/fabrique_de_gif/start?rev=1604962646**
Article mis à jour: **2020/11/09 23:57**