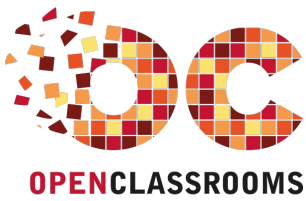
Créer des GIF animés

Par Phoenix76



www.openclassrooms.com

Dernière mise à jour le 4/07/2010

Sommaire

Créer des GIF animés

Par  Phoenix76

Mise à jour : 04/07/2010
Difficulté : Intermédiaire Durée d'étude : 1 heure

La librairie GD offre de nombreuses possibilités pour manipuler vos images. Cependant, une manque cruellement : l'animation. Pourtant, vous aimeriez bien ajouter de petits effets à vos images. Mais vous avez eu beau chercher une solution, rien de bien concluant.
Vous croyiez que tout était perdu ? Eh bien ~~oui~~ erreur : Phoenix76 est là et, ensemble, nous allons pouvoir défiger GD.

Prêts ? À l'attaaaaque ! 

Pour bien comprendre ce tutoriel, vous devez avoir lu l'ensemble du tutoriel officiel sur le PHP.
Il y aura quelques notions que vous n'aurez certainement pas vues, mais pas de panique : elles seront toutes expliquées.

Sommaire du tutoriel :

Les bases

C'est quoi, un GIF animé ?

Le GIF animé est une amélioration du format GIF, apparue en 1989. Il permet de stocker une succession d'images auxquelles est associé un temps d'affichage. Autrement dit, il permet d'obtenir une animation.

La classe magique

GD permet de créer des GIF simples : donc, si l'on suit la définition, il suffirait de « coller » plusieurs GIF entre eux pour obtenir une belle animation. Malheureusement, regrouper des images entre elles n'est pas chose facile, GD en est même incapable. Mais n'ayez crainte, **László Zsidi** a développé une classe comblant cette petite lacune : **GIFEncoder** (rendez-vous ici pour la télécharger).

Une classe permet de créer des objets contenant un ensemble de fonctions (les méthodes) et de variables (les attributs).
Ici la classe GIFEncoder permet de créer des objets « GIF animés » avec lesquels on peut interagir, via des méthodes comme GetAnimation() pour obtenir notre animation.
Si vous voulez en savoir plus sur la POO et tout ce qui en découle (les classes, les méthodes...), direction Wikipédia.

Pour coller les GIF entre eux, GIFEncoder effectue des opérations binaires complexes sur les images. Nous n'étudierons pas cela ici, mais voici un très bon lien traitant de ce sujet.

Nous nous intéresserons seulement à l'utilisation de deux méthodes (GIFEncoder et GetAnimation) car ce sont les seules à destination de l'utilisateur.

Squelette du script

Vous l'aurez sûrement deviné : dans cette partie, nous établirons une base pour vos futures créations.

Tout d'abord, on va importer notre classe :

Code : PHP

```php
<?php
require('libs/gif.class.php');
?>
```

J'ai l'habitude de placer mes classes dans un dossier libs ; vous n'êtes pas obligés de le faire, le principal est que vous puissiez vous y retrouver.

Bon, maintenant, on reprend notre définition : *« une succession d'images auxquelles sont associé un temps d'affichage »*.
Comment peut-on traduire ça en PHP ? Avec des tableaux, pardi.

Code : PHP

```php
<?php
$animation = array(); //Chaque élément de ce tableau sera une
image...
$duree = array(); //...avec une durée associée
?>
```

On peut maintenant créer nos images :

Code : PHP

```php
<?php
define('LARGEUR', X);
define('HAUTEUR', Y);

$image = imagecreate(LARGEUR, HAUTEUR);
//À vous de jouer avec les couleurs, les formes...
//On place l'image dans notre tableau animation
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = Z; //Attention, le temps d'affichage des images s'écrit
en centième de seconde, 1s = 100cs

//On détruit la dernière image pour libérer de la mémoire
imagedestroy($image);
?>
```

Vous vous demandez sûrement à quoi servent les fonctions ob_start et ob_get_clean. Plutôt que de vous l'expliquer maintenant dans un long discours abstrait, nous verrons leur rôle dans la partie pratique. Retenez pour l'instant que les trois lignes surlignées placent notre image dans le tableau animation.

Bon, ça avance bien, on a deux tableaux avec nos images et durées d'affichage. On n'a plus qu'à tout coller.
Pour cela, on va utiliser la méthode GIFEncoder.

Voici sa signature et la liste des paramètres :

Code : Autre

```
GIFEncoder(array $tab, array $time, int $loops, int $disposal, int $r, int $g, i

$tab : Les images

$time : Durées associées à chaque image

$loops : L'animation est jouée n fois, 0 <=> infini

$disposal : Si quelqu'un trouve à quoi ça sert...

$r, $g, $b : Permettent d'ajouter de la couleur à des GIF transparents suivant l

$type :
        'bin' -> $tab contient des images générées par gd
        'url' -> $tab contient des url vers des fichiers de type image
```

Ces paramètres sont tous obligatoires, voici la méthode par défaut :

Code : PHP

```php
<?php
$gif = new GIFEncoder($animation, $duree, 0, 2, 0, 0, 0, 'bin');
?>
```

L'objet gif contient désormais notre animation. Il nous reste à afficher l'image ou à l'enregistrer. Pour ce faire, on va utiliser une autre méthode de notre classe : GetAnimation. Elle retourne notre image sous forme de **chaîne de caractères**.

Donc, pour afficher, on fait :

Code : PHP

```php
<?php
header('Content-type: image/gif'); //Indique au navigateur que l'on
veut afficher une image GIF et non pas des caractères
echo $gif->GetAnimation();
?>
```

Et pour enregistrer :

Code : PHP

```php
<?php
$fichier = fopen(IMAGE, 'w+');
fputs($fichier, $gif->GetAnimation());
fclose($fichier);
?>
```

Notre squelette finalisé

Code : PHP

```php
<?php
require('libs/gif.class.php');

define('LARGEUR', X);
define('HAUTEUR', Y);

$animation = array(); //Chaque élément de ce tableau sera une image...
$duree = array(); //...avec une durée associée

//On crée les différentes images de l'animation et on définit leur temps d'affichage
$image = imagecreate(LARGEUR, HAUTEUR);
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = Z; //Attention, le temps d'affichage des images s'écrit en centième de seconde, 1s = 100cs.

//On détruit la dernière image pour libérer de la mémoire
imagedestroy($image);

//On crée le GIF à partir de l'animation.
$gif = new GIFEncoder($animation, $duree, 0, 2, 0, 0, 0, 'bin');

//On affiche
header('Content-type: image/gif'); //Indique au navigateur qu'on veut afficher une image GIF et non pas des caractères
echo $gif->GetAnimation();

//Ou on enregistre
$fichier = fopen(IMAGE, 'w+');
fputs($fichier, $gif->GetAnimation());
fclose($fichier);
?>
```

Exemples

Une bannière

Objectif : Appliquer le script précédent.

Aperçu : 

Comme vous pouvez le constater, cette animation est ~~moche, inutile et réalisable en 10s~~ composée de deux images : l'une de fond rouge, l'autre de fond bleu avec un texte identique de couleur blanche. Le temps d'affichage des images est de 1 seconde.

Première chose à faire : on reprend notre squelette.
Ensuite, on définit un message et des dimensions qui lui sont adaptées :

Code : PHP

```php
<?php
define('LARGEUR', 210);
define('HAUTEUR', 25);
$message = 'Mon premier gif en php';
?>
```

On peut maintenant passer à la création de la première image :

Code : PHP

```php
<?php
$image = imagecreate(LARGEUR, HAUTEUR);
$fond = imagecolorallocate($image, 255, 0, 0);
$blanc = imagecolorallocate($image, 255, 255, 255);
imagestring($image, 5, 5, 5, $message, $blanc);
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = 100; //Attention, le temps d'affichage des images s'écrit
en centième de seconde, 1s = 100cs
imagedestroy($image);
?>
```

Si vous ne comprenez pas tout, allez revoir le cours sur GD.
Je vous laisse le soin de créer la deuxième image.
Vous avez fini ? Bien : vous devriez avoir quelque chose qui ressemble à ceci :

Secret (cliquez pour afficher)

Code : PHP

```php
<?php
require('libs/gif.class.php');
$animation = array();
$duree = array();

define('LARGEUR', 210);
define('HAUTEUR', 25);
$message = 'Mon premier gif en php';

$blanc = imagecolorallocate($image, 255, 255, 255);

$image = imagecreate(LARGEUR, HAUTEUR);
$fond = imagecolorallocate($image, 255, 0, 0);
imagestring($image, 5, 5, 5, $message, $blanc);
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = 100;
imagedestroy($image);

$image = imagecreate(LARGEUR, HAUTEUR);
$fond = imagecolorallocate($image, 0, 0, 255);
imagestring($image, 5, 5, 5, $message, $blanc);
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = 100;
imagedestroy($image);

$gif = new GIFEncoder($animation, $duree, 0, 2, 0, 0, 0, 'bin');
header ("Content-type: image/gif");
echo $gif->GetAnimation();
?>
```

Et voilà : normalement, une belle image s'affiche sous vos yeux ébahis. Nous pouvons maintenant passer à la seconde partie.

Hé minute, tu ne nous as toujours pas expliqué à quoi servait ob_start et ob_get_clean !

Ah oui, quel étourdi !
Bon, comme l'apprentissage passe par la pratique, enlevez-moi ces deux fonctions :

Code : PHP

```php
<?php
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
?>
```

Cela devient :

Code : PHP

```php
<?php
$animation[] = imagegif($image);
?>
```

Actualisez votre page et... un tas de caractères bizarres apparaissent. Ne vous inquiétez pas, c'est normal.
Lorsque vous exécutez la fonction imagegif, celle-ci crée l'image voulue, l'affiche sous forme d'une chaîne de caractères et retourne un booléen pour savoir si tout s'est bien passé.
Ainsi, dans le deuxième code, vous affichez votre image (les caractères louches) et vous créez un tableau de booléen. C'est bien, mais nous, on veut récupérer la chaîne pour l'insérer dans le tableau.

Dans cet objectif, on va utiliser un tampon. Cette chose énigmatique cache un fabuleux pouvoir d'absorption. En effet, de son ouverture à sa fermeture, il capte une quantité incroyable de flux de toutes les couleurs, mais sans jamais les afficher.
La fonction ob_start crée le tampon. Puis on exécute imagegif ; le tampon enregistre la chaîne qu'elle affiche (notre image). Enfin, on vide le tampon dans notre tableau.
Bon, je ne vous fais pas de dessin, j'espère avoir été suffisamment clair.

Exercices supplémentaires (par ordre de difficulté croissante)

- Au lieu de créer des images, importez-les (utilisez le paramètre 'url').
- Ajoutez une image avec un fond vert et un texte en orange.
- Donnez un texte différent à chaque image en adaptant sa taille.
- Déplacez votre texte d'un coin à un autre.

Un flux RSS en image

Objectif : réaliser une image pour indiquer sur les forums les dernières news de son site.

Aperçu :

Pour cette partie, je vais vous donner quelques indications et ce sera à vous de jouer.
Tout d'abord, récupérez l'image d'arrière-plan (généreusement fournie par Jerry Wham), ou créez votre propre image.
Pour que tout le monde ait la même base, créez une nouvelle table :

Code : SQL

```sql
CREATE TABLE IF NOT EXISTS `news_gif` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `titre` varchar(100) NOT NULL,
  `timestamp` int(11) NOT NULL,
  PRIMARY KEY  (`id`),
  KEY `id` (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=4 ;

INSERT INTO `news_gif` (`id`, `titre`, `timestamp`) VALUES
(1, 'Ouverture du site', 1171298713),
(2, 'Ouverture forum', 1172191293),
(3, 'Test fil rss', 1203603483);
```

Pour l'effet « machine à écrire », vous aurez besoin de la fonction str_split.
Si vous n'avez pas php5, vous pouvez la remplacer par : preg_split('//', string, -1, PREG_SPLIT_NO_EMPTY).

N'oubliez pas d'indiquer la date de la dernière modification.

Placez votre code dans une fonction car, à chaque ajout / modification / suppression de news, il faudra mettre l'image à jour, et ce serait bête de réécrire trois fois le même code.

Je vous ai tout dit : à vous de jouer.

Correction :

Secret (cliquez pour afficher)

Code : PHP

```php
<?php
function creer_gif_rss()
{
        require('libs/connexion.php'); //Fichier de connexion à
la bdd
        require('libs/gif.class.php');

        $animation = array();
        $duree = array();

        $retour = mysql_query('SELECT titre, timestamp FROM
news_gif ORDER BY id DESC LIMIT 0, 1') or die(mysql_error());
//Sélectionne le dernier article
        $donnees = mysql_fetch_array($retour);

 /* La chaîne correspondra au dernier titre du blog.
* L'espace blanc crée une image vide afin de rendre l'animation
plus fluide (essayez de l'enlever).
*/

        $chaine = ' '.$donnees['titre'];
        $taille_chaine = strlen($chaine);
        $chaine = str_split($chaine); //Transforme la chaîne en
tableau.

        $image = imagecreatefrompng('fond_rss.png'); //Notre joli
fond :p
```

```
        $fond = imagecolorallocate($image, 249, 249, 249);
        $bleu = imagecolorallocate($image, 0, 0, 255);
        $noir = imagecolorallocate($image, 9, 9, 9);

        imagestring($image, 5, 120, 5, "Bob's home", $bleu); //Un
petit titre
        imagestring($image, 3, 70, 40, 'Le '.date('d/m/Y \a H\hi',
$donnees['timestamp']), $noir); //Affiche la date de la dernière
modification sur l'article

        foreach($chaine as $i => $carac) //Pour chaque lettre de
la chaîne, on crée une nouvelle image
        {
                $txt .= $carac; //On affiche jusqu'au caractère
actuel
                imagestring($image, 4, 60, 23, $txt, $noir);

                ob_start(); //Je ne vous explique pas : vous
devriez avoir compris ^^
                imagegif($image);
                $animation[] = ob_get_clean();

  if($i == 0) //Pause lorsque rien ne s'est affiché (plus
esthétique)
   $duree[] = 45;
  else if($i+1 == $taille_chaine) //Si on est à la fin de la
chaîne, on marque une pause
                        $duree[] = 200;
                else
                        $duree[] = 15;
        }
        imagedestroy($image); //On détruit l'image pour libérer
de la mémoire

        $gif = new GIFEncoder($animation, $duree, 0, 2, 0, 0, 0,
'bin'); //On génère l'animation
        $fichier = fopen(IMAGE, 'w+');
        fputs($fichier, $gif->GetAnimation()); //On enregistre
        fclose($fichier);
}
?>
```

Et voilà, notre fonction est terminée ; vous pouvez vérifier qu'elle effectue bien son travail en éditant votre table. 

Exercices supplémentaires (par ordre de difficulté croissante)

- Écrivez le titre de la droite vers la gauche.
- Faites varier la durée entre l'affichage des différentes lettres pour améliorer l'effet « machine à écrire ».
- Créez un effet de disparition lorsque le titre est affiché en entier.
- Modifiez la fonction pour pouvoir afficher les trois dernières news et leur date de création.

Q.C.M.

Le premier QCM de ce cours vous est offert en libre accès.
Pour accéder aux suivants

Connectez-vous Inscrivez-vous

La librairie GD ne permet pas de :

- ◯ créer des GIF.
- ◯ modifier une image.
- ◯ coller des GIF entre eux.

Qu'est-ce qu'une animation ?

- ◯ Une image avec un temps d'affichage variable.
- ◯ Une tripotée d'images avec une durée d'affichage variable.
- ◯ Une succession d'images avec un temps d'affichage constant.

Que fait ce code ?

Code : PHP

```php
<?php
require('libs/gif.class.php');

define('LARGEUR', 210);
define('HAUTEUR', 25);

$animation = array();
$duree = array();
$message = 'Mon premier gif en php';

$image = imagecreate(LARGEUR, HAUTEUR);
$fond = imagecolorallocate($image, 255, 0, 0);
$blanc = imagecolorallocate($image, 255, 255, 255);
imagestring($image, 5, 5, 5, $message, $blanc);
ob_start();
imagegif($image);
$animation[] = ob_get_clean();
$duree[] = 100;
imagedestroy($image);

$gif = new GIFEncoder($animation, $duree, 0, 2, 0, 0, 0, "bin");
echo $gif->GetAnimation();
?>
```

- ◯ Il plante.
- ◯ Il affiche une animation.
- ◯ Il écrit un tas de trucs bizarres.

Test ultime : combien y a-t-il d'erreurs dans ce script ?

Code : PHP

```php
<?php
require('libs/connexion.php'); //On se connecte à notre BDD
require('libs/gif.class.php');

$retour = mysql_query('SELECT titre, timestamp FROM news_gif ORDER
BY id DESC LIMIT 0, 1'); //Sélectionne le dernier article
$donnees = mysql_fetch_array($retour);
$chaine = $donnees['titre']; //La chaîne correspondra au dernier
titre du blog
$taille_chaine = strlen($chaine);
str_split($chaine); //Transforme la chaîne en tableau

$animation = array();
$duree = array();

$largeur = 220;
$hauteur = 60;
$image = imagecreatefromjpeg('fond_rss.jpg'); //Notre joli fond

$fond = imagecolorallocate($image, 249, 249, 249);
$bleu = imagecolorallocate($image, 0, 0, 255);
$noir = imagecolorallocate($image, 9, 9, 9);
$vert = imagecolorallocate($image, 146, 201, 146);
```

```
imagestring($image, 5, 65, 5, "Bob's home", $bleu); //Un petit titre
imagestring($image, 3, 9, 40, 'Le '.date('d/m/Y \a H\hi',
$donnees['timestamp']), $vert); //Affiche la date de la dernière
modification sur l'article

foreach($chaine as $carac) //Pour chaque lettre de la chaîne, on
crée une nouvelle image
{
        $txt = $carac; //On affiche jusqu'au caractère actuel
        imagestring($image, 4, 9, 23, $txt, $noir);

        ob_start(); //Je ne vous explique pas : vous devriez avoir
compris
        imagegif($image);
        $animation = ob_get_clean();

        if($i+1 == $taille_chaine) //Si on est à la fin de la
chaîne, on marque une pause
                $duree[] = 200;
        else
                $duree[] = 15;
}
imagedestroy($image); //On détruit l'image pour libérer de la
mémoire

$gif = new GIFEncoder($duree, $animation, 0, 2, 0, 0, 0, "bin");
//On génère l'animation
$fichier = fopen('image.gif', 'w');
fputs($fichier, $gif->GetAnimation()); //On enregistre
fclose($fichier);
?>
```

- ◯ 5.
- ◯ 6.
- ◯ 7.

Correction !

Statistiques de réponses au QCM

Vous voici arrivés à la fin de ce tutoriel : vous êtes désormais capables de créer de beaux GIF animés. Attention ! **Restez dans la sobriété** : c'est toujours très désagréable d'assister à un feu d'artifice sur une page Web.

Ce tutoriel a été corrigé par les zCorrecteurs.