

## Problème 1 : Non fonctionnement du changement de photos dans la modal

Lorsque l'utilisateur cliquait sur les flèches de navigation (".mg-prev" et ".mg-next") dans la lightbox, l'image affichée ne changeait pas.

### 1. Délégation d'Événements

Avant (avec problème) :

```
$(".gallery").on("click", ".mg-prev", () =>
$.fn.mauGallery.methods.prevImage(options.lightboxId) );
$(".gallery").on("click", ".mg-next", () =>
$.fn.mauGallery.methods.nextImage(options.lightboxId) );
```

Après (problème réglé) :

```
$(".gallery").on("click", ".mg-prev", function() {
$.fn.mauGallery.methods.prevImage(options.lightboxId);
});
$(".gallery").on("click", ".mg-next", function() {
$.fn.mauGallery.methods.nextImage(options.lightboxId);
});
```

**Explication** : La délégation d'événements assure que les événements sont correctement capturés même si les éléments sont ajoutés dynamiquement après le chargement initial du DOM.

### 2. Identification de l'Image Active

Avant (avec problème) :

```
let activeImage = null;
$(".img.gallery-item").each(function() {
if ($(this).attr("src") === $(".lightboxImage").attr("src")) {
activeImage = $(this);
}
});
```

Après (problème réglé) :

```
const lightboxImageSrc =
$('#${lightboxId}`).find(".lightboxImage").attr("src");
let activeImage = null;
$(".img.gallery-item").each(function() {
if ($(this).attr("src") === lightboxImageSrc) {
activeImage = $(this);
}
});
```

**Explication** : L'utilisation d'une variable pour stocker le src de l'image affichée dans la lightbox améliore la robustesse de la sélection de l'image active.

### 3. Collection d'Images Basée sur le Tag Actif

**Avant (avec problème) :**

```
let activeTag = $(".tags-bar span.active-tag").data("images-toggle");
let imagesCollection = [];
if (activeTag === "all") {
    $(".item-column").each(function() {
        if ($(this).children("img").length) {
            imagesCollection.push($(this).children("img"));
        }
    });
} else {
    $(".item-column").each(function() {
        if (
            $(this)
                .children("img")
                .data("gallery-tag") === activeTag
        ) {
            imagesCollection.push($(this).children("img"));
        }
    });
}
```

**Après (problème réglé) :**

```
let imagesCollection = [];
if (activeTag === "all") {
    $(".item-column").each(function() {
        if ($(this).children("img").length) {
            imagesCollection.push($(this).children("img"));
        }
    });
} else {
    $(".item-column").each(function() {
        if ($(this).children("img").data("gallery-tag") === activeTag) {
            imagesCollection.push($(this).children("img"));
        }
    });
}
```

**Explication** : La logique a été simplifiée pour s'assurer que les images sont correctement sélectionnées en fonction du tag actif.

#### 4. Calcul de l'Index de l'Image Suivante/Précédente

Avant (avec problème) :

```
let index = 0,
    next = null;

$(imagesCollection).each(function(i) {
    if ($(activeImage).attr("src") === $(this).attr("src")) {
        index = i ;
    }
});
next =
    imagesCollection[index] ||
    imagesCollection[imagesCollection.length - 1];
$(".lightboxImage").attr("src", $(next).attr("src"));
```

Après (problème réglé) :

```
Copier le code
let index = imagesCollection.findIndex(img => img.attr("src") ===
activeImage.attr("src"));
let nextIndex = (index + 1) % imagesCollection.length; // pour l'image suivante
let prevIndex = (index - 1 + imagesCollection.length) % imagesCollection.length; //
pour l'image précédente
$(".lightboxImage").attr("src", imagesCollection[nextIndex].attr("src"));
```

**Explication** : L'utilisation de `findIndex` pour trouver l'index de l'image active et le calcul circulaire de l'index de l'image suivante/précédente assurent une navigation correcte et robuste entre les images.

**Problème 2 : Appliquer le background sur le filtre actif uniquement.**

Avant (avec problème) :

```
$(".active-tag").removeClass("active-tag");
```

Après (problème réglé) :

```
$(".nav-link").removeClass("active-tag");
```

Ajout de cet élément CSS

```
.active-tag {
    background-color: #5E5926 !important;
```

```
color: white !important;
}
```

Suppression de cette partie du bootstrap.css

```
.nav-pills .nav-link.active,  
.nav-pills .show > .nav-link {  
  color: #fff;  
  background-color: #0d6efd;  
}
```

**Explication** : Le changement de `.active-tag` par `.nav-link` permet de correctement détecter l'élément à supprimer. Il fallait ensuite créer un style pour la class `active-tag` et supprimer le style `".nav-pills .nav-link.active, .nav-pills .show > .nav-link"` car sinon le bouton "tous" restait avec un background.