



Rapport du projet

Sujet: «Filter maker»

Groupe:

BERRAI Dyhia	15608005
HUARD Alicia	15601834

L3: TSI 2020 M. BOUBCHIR

Nous avons choisi de faire notre projet sur les filtres appliqués à des images (640x640). Nous avons donc codés les fonctions suivantes:

1. Load_Image()
2. Original()
3. Save()
4. Zoom_in()
5. Zoom_out()
6. Grey_filter()
7. Sobel_filter()
8. Mirror()
9. Negative_filter()
10. Close()

Interface Graphique de l'application

Pour avoir une interface nous avons utilisés l'outil ***guilbuilder***, qui permet aux utilisateurs de **Scilab** de créer des interfaces graphiques plus facilement et plus rapidement. Elle permet aussi de générer le code de celle-ci.

Voici les fenêtres d'utilisation de ***guilbuilder***:

La première permet la création des objets (boutons, etc.)

La deuxième permet d'avoir un aperçu de la future application

La troisième quant à elle permet d'ajuster les objets de la fenêtre, pour que celle-ci ai un aspect plus agréable.

Implémentions des fonctions

1. Load_Image()

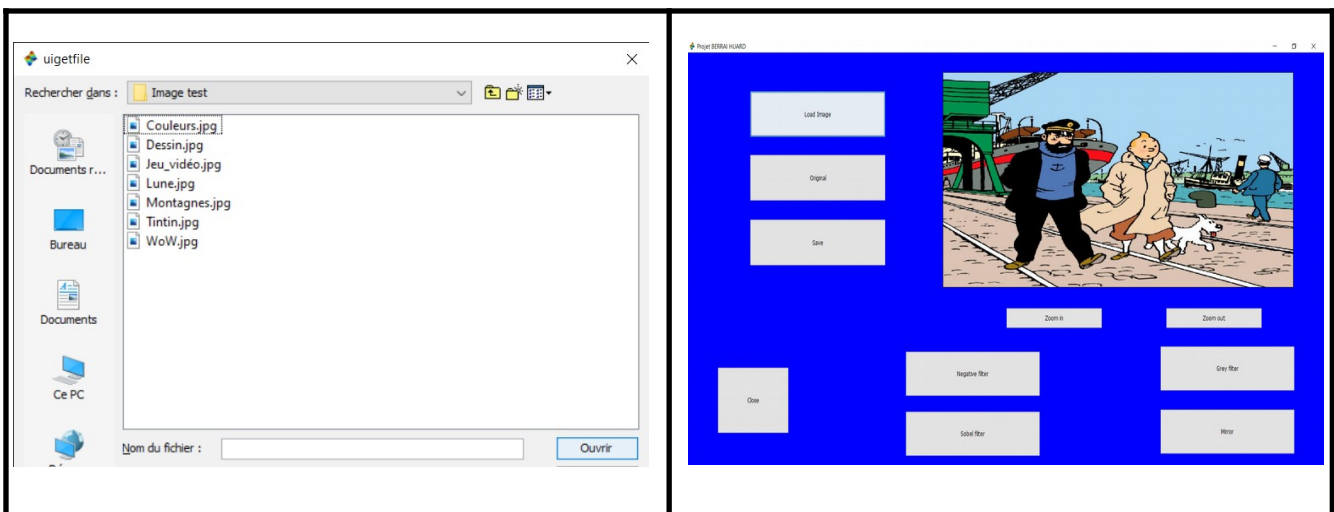
Comme son nom l'indique, cette fonction permet de charger une image (640x640) dans l'interface graphique.

```
function Load_Image_callback(handles)

    filePath = uigetfile('*');
    image = imread(filePath);
    imshow(image);
    handles.OriginalImage = image;
    handles.CopyImage      = image;
    handles.MirrorImage     = 0;
    handles = resume(handles);

endfunction
```

Exemple d'utilisation:



2. Original()

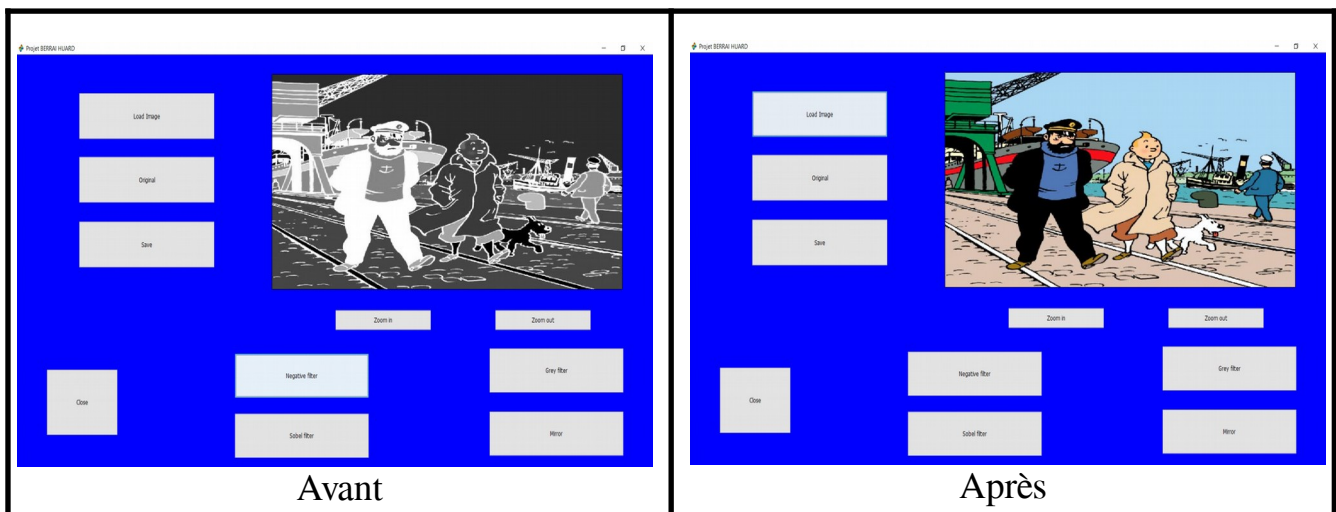
Cette fonction permet de retourner à l'image d'origine.

```
function Original_callback(handles)

    image = handles.OriginalImage;
    imshow(image);
    handles.CopyImage= image;
    handles = resume(handles);

endfunction
```

Exemple d'utilisation:



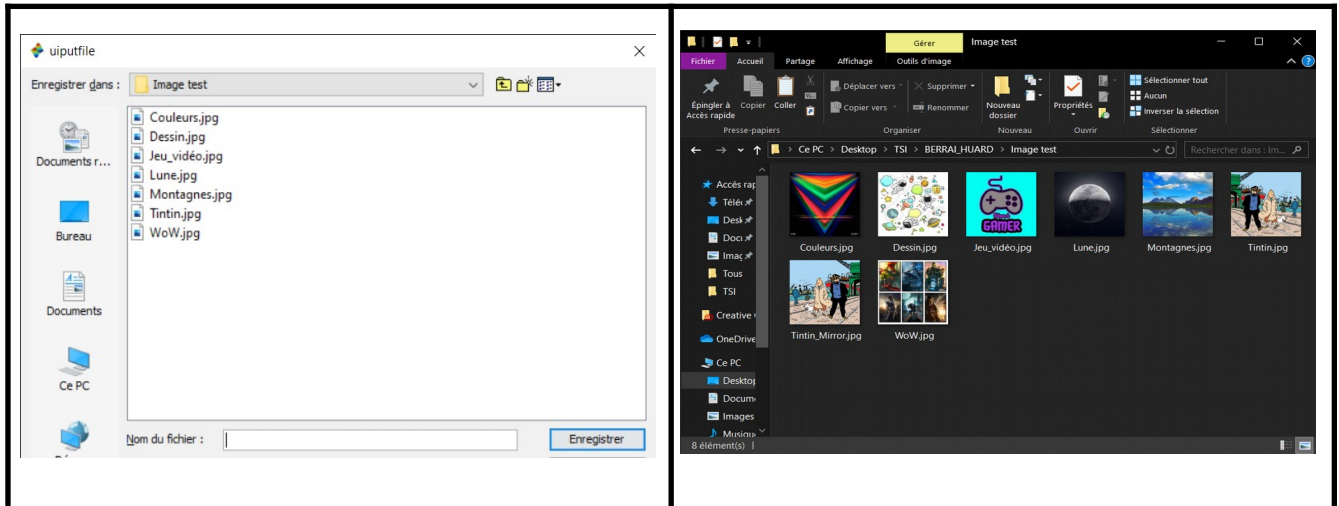
3. Save()

Cette fonction permet de sauvegarder la nouvelle image et de lui donner un nom.

```
function Save_callback(handles)
    imageTosave=handles.CopyImage;
    imshow(imageTosave);
    [fileName,filePath]=uiputfile(["*.jpeg"; "*.png"; "*.jpg"]);
    fileName=fullfile(filePath,fileName);
    imwrite(imageTosave,string(fileName+".jpg"));

endfunction
```

Exemple d'utilisation:



4. Zoom_in()

Cette fonction permet de sélectionner une partie d'une image et de faire un zoom dessus.

```
function Zoom_in_callback(handles)

    handles.BeforeZoom=handles.CopyImage;
    image = handles.CopyImage;
    rect1=rubberbox();
    imheight=size(image,"r");

    rect1(2)=imheight-rect1(2);
    zoomed=imcrop(image,rect1);
    imshow(zoomed);
    height=size(zoomed,"r");
    handles.CopyImage=zoomed;
    handles = resume(handles);

endfunction
```

Exemple d'utilisation:



5. Zoom_out()

Cette fonction quant à elle retourne à l'image d'origine ce qui permet de retirer l'effet du zoom.

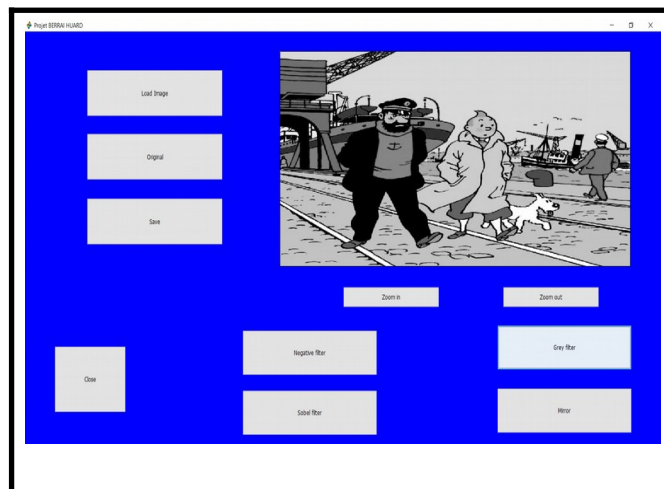
```
function Zoom_out_callback(handles)
    imshow(handles.BeforeZoom);
    if handles.CopyImage==handles.BeforeZoom then
        handles.BeforeZoom=handles.OriginalImage;
    end
    handles.CopyImage=handles.BeforeZoom;
    handles = resume(handles);
endfunction
```

6. Grey_filter()

Cette fonction permet d'appliquer un filtre gris aux pixels RGB de l'image.

```
function Grey_filter_callback(handles)
    image = rgb2gray(handles.CopyImage);
    imshow(image);
    handles.CopyImage = image;
    handles = resume(handles);
endfunction
```

Exemple d'utilisation:

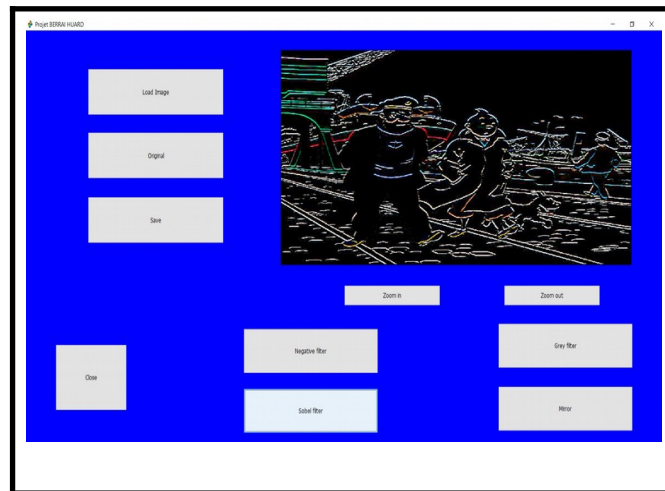


7. Sobel_filter()

Cette fonction permet d'appliquer un filtre Sobel à l'image.

```
function Sobel_filter_callback(handles)
    image = handles.CopyImage;
    filter = fspecial('sobel');
    image = imfilter(image, filter);
    imshow(image);
    handles.CopyImage = image;
    handles = resume(handles);
endfunction
```

Exemple d'utilisation:

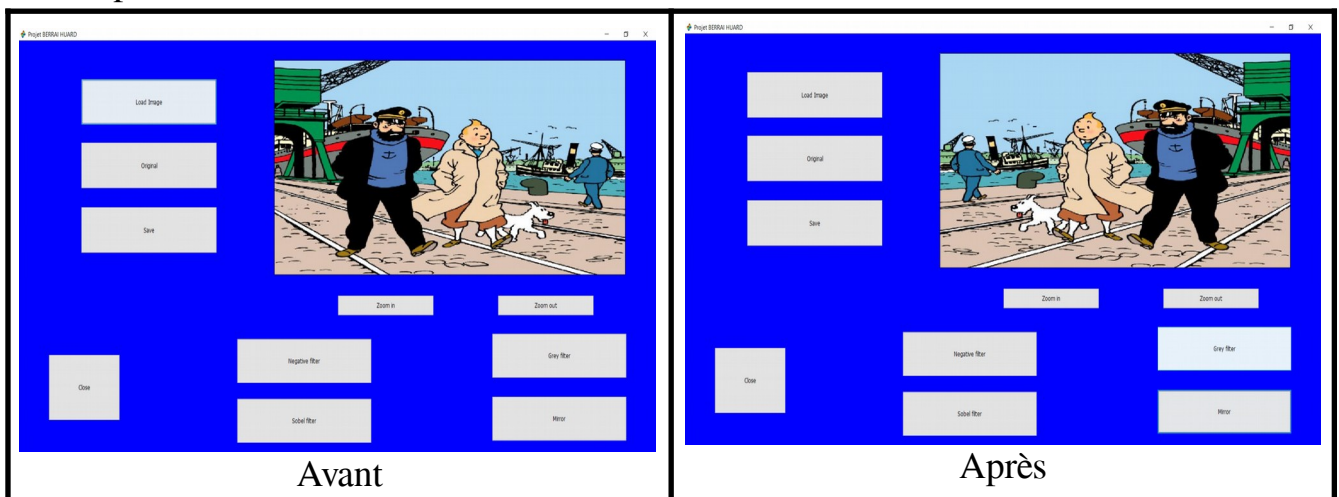


8. Mirror()

Cette fonction permet d'inverser l'image pour avoir l'effet miroir.

```
function Mirror_callback(handles)
    test=~handles.MirrorImage;
    if test then
        image = handles.CopyImage(:, $:-1:1, :);
    else
        image = handles.OriginalImage;
    end
    imshow(image);
    handles.MirrorImage=test;
    handles.CopyImage = image;
    handles = resume(handles);
endfunction
```

Exemple d'utilisation:



9. Negative_filter()

Cette fonction permet d'appliquer un filtre négatif à l'image.

```
function Negative_filter_callback(handles)
    [n1, nc] = size(handles.CopyImage);
    image = handles.CopyImage;
    for i = 1:n1
        for j = 1:nc
            image(i, j) = 255 - image(i, j)
        end;
    end
    imshow(image);
    handles.CopyImage = image;
    handles = resume(handles);
endfunction
```

Exemple d'utilisation:

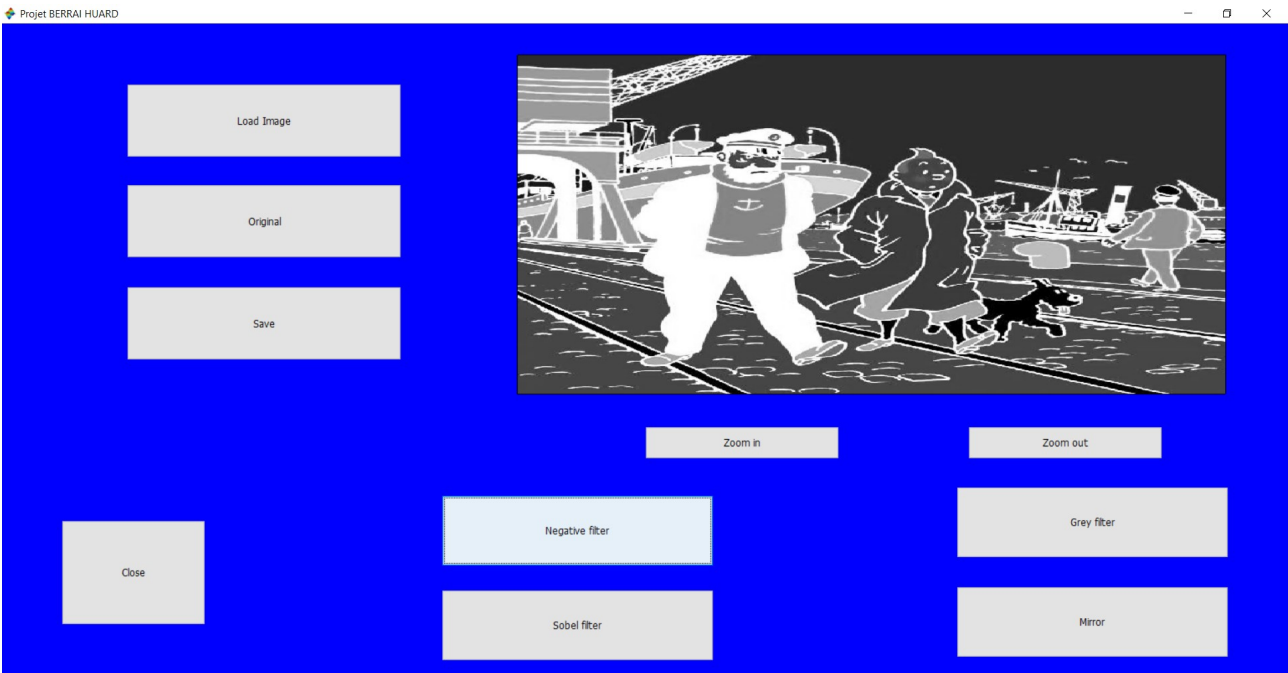


10. Close()

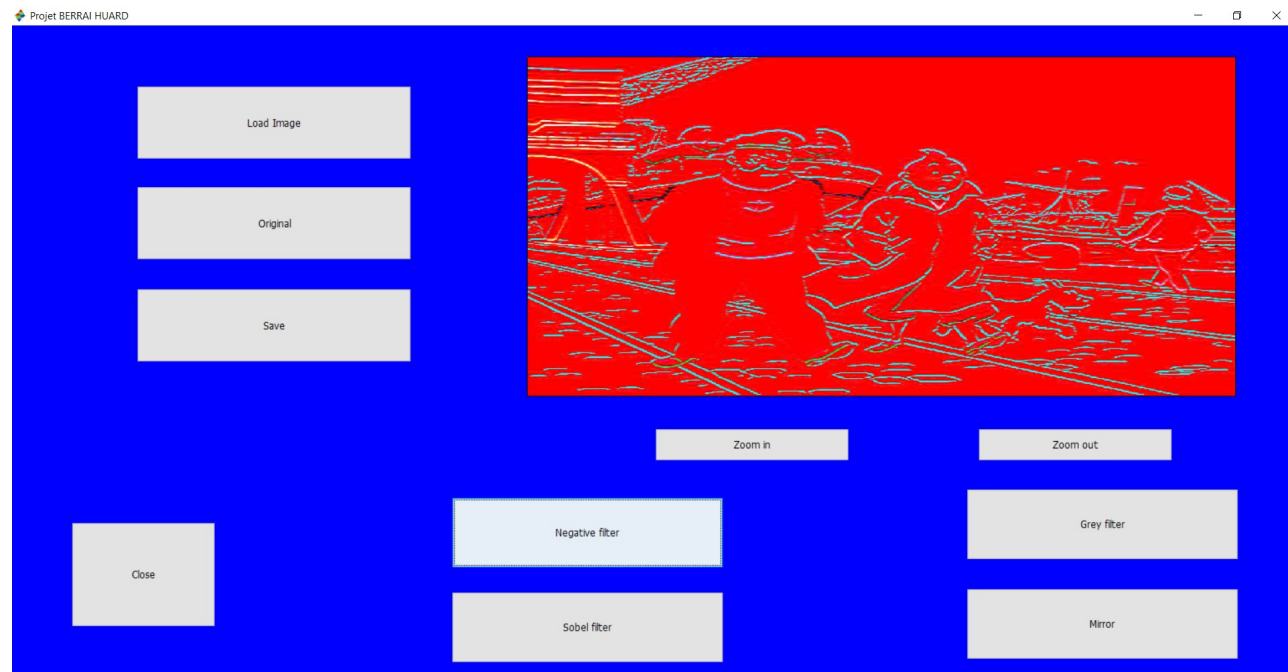
Cette fonction permet de fermer la fenêtre.

```
function Close_callback(handles)
    close(f);
endfunction
```

D'autres Images



Filtre Gris + Filtre Négatif



Filtre Sobel + Filtre Négatif