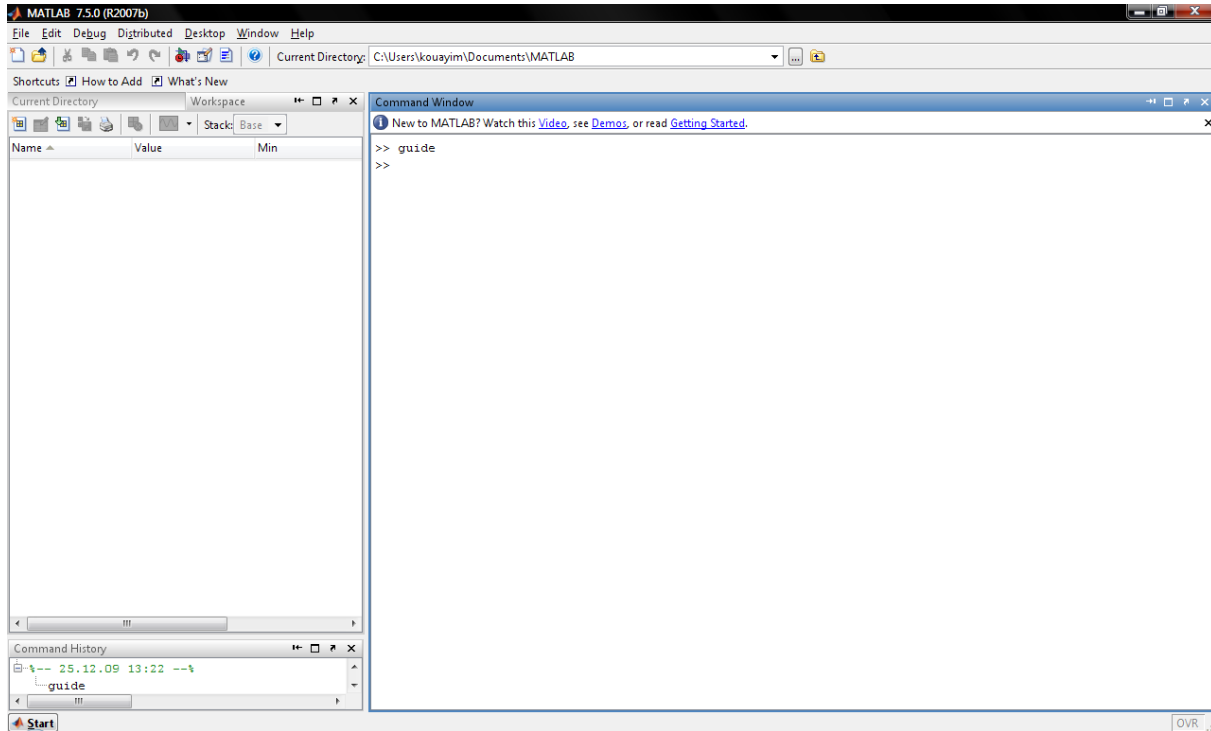


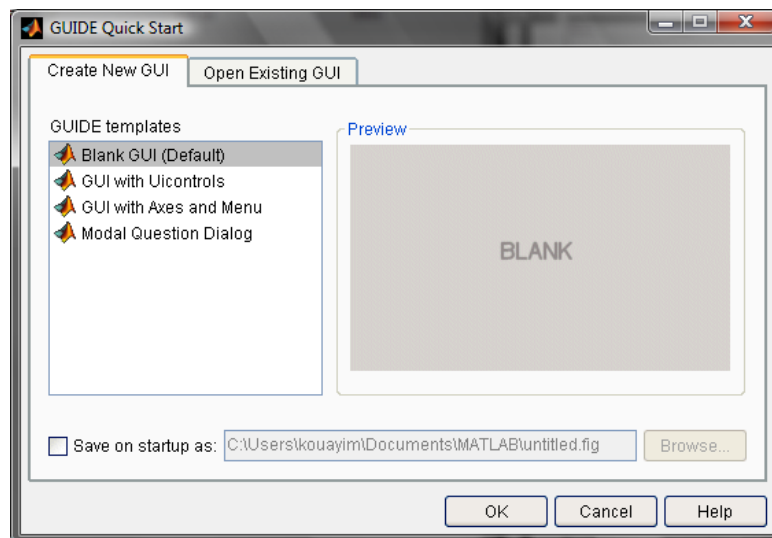
## Solution Proposée par Georges Bonga

### Matlab-Exo 1, soumis par Hadil:( GUI avec Guide)

Pour créer un GUI avec « Guide », il faut commencer par ouvrir «Guide ». On le fait en entrant l'instruction « guide » dans la fenêtre de commande de Matlab.



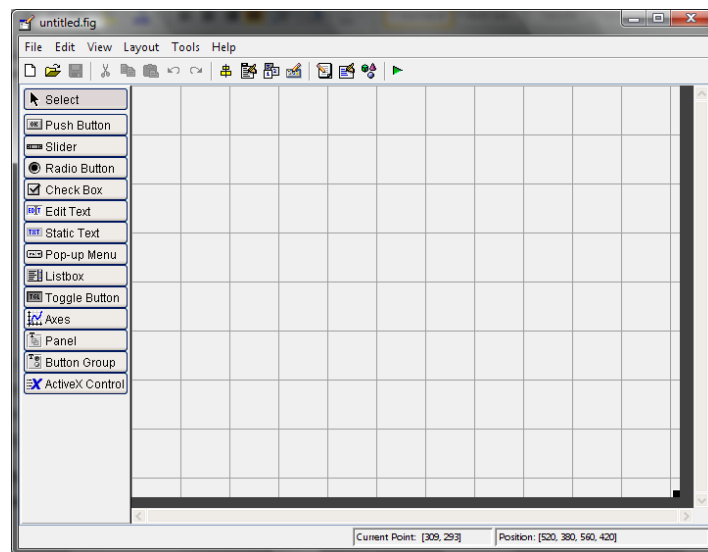
Lorsque on entre cette instruction, une fenêtre s'ouvre, dans laquelle on doit sélectionner quel genre de GUI on veut avoir comme GUI de départ.



Il est possible de choisir des GUI qui ont déjà certain éléments. Dans notre cas, nous allons prendre un GUI vide (Blank GUI), qui est le GUI par défaut. Dans cette fenêtre il est aussi possible de sélectionner où sera enregistré notre GUI. Par défaut il est enregistré dans le répertoire de Matlab.

## Matlab-Exo 1

Quand on clique sur OK, notre future GUI. Pour le moment il est encore vide et est nommé par défaut « untitled.fig ».



Maintenant nous devons ajouter les différents éléments dont nous avons besoin.

Notre GUI doit nous servir à ouvrir des photos, ce qui veut dire que nous aurons besoin d'axes pour les afficher. À la marge nous allons donc sélectionner l'élément « axes » et l'ajouté à notre GUI.

Pour ouvrir une photo nous allons devoir la sélectionner, ce qui veut dire que nous avons besoin d'un bouton que nous nommerons « Open ». Notre bouton « Open » sera un « Push Button ». Nous sélectionnons donc un « Push Button » à la marge et nous l'ajoutons à notre GUI.

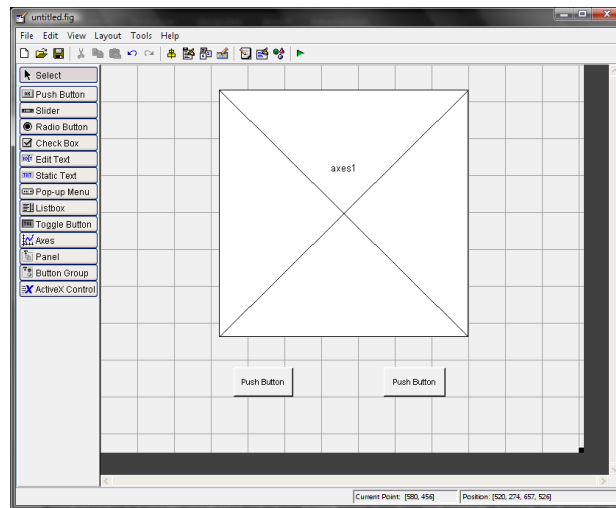
Nous supposons que les photos vont être l'objet d'un traitement par d'autres fonction, et nous supposons que les résultats vont être enregistrés, c'est pourquoi nous allons ajouter un bouton pour l'enregistrement.

Pour l'enregistrement, plusieurs possibilités s'offrent à nous.

- Nous pouvons par exemple faire en sorte que notre GUI enregistre directement les données ouvertes dans un dossier prédéfini.
- Nous pouvons ajouter un bouton « Save » sur lequel l'utilisateur doit cliquer avant que les photos ne soient enregistrées dans un dossier prédéfini.
- Nous pouvons ajouter un bouton « Save » sur lequel l'utilisateur doit cliquer et sélectionner le dossier dans lequel les données doivent être enregistrées.

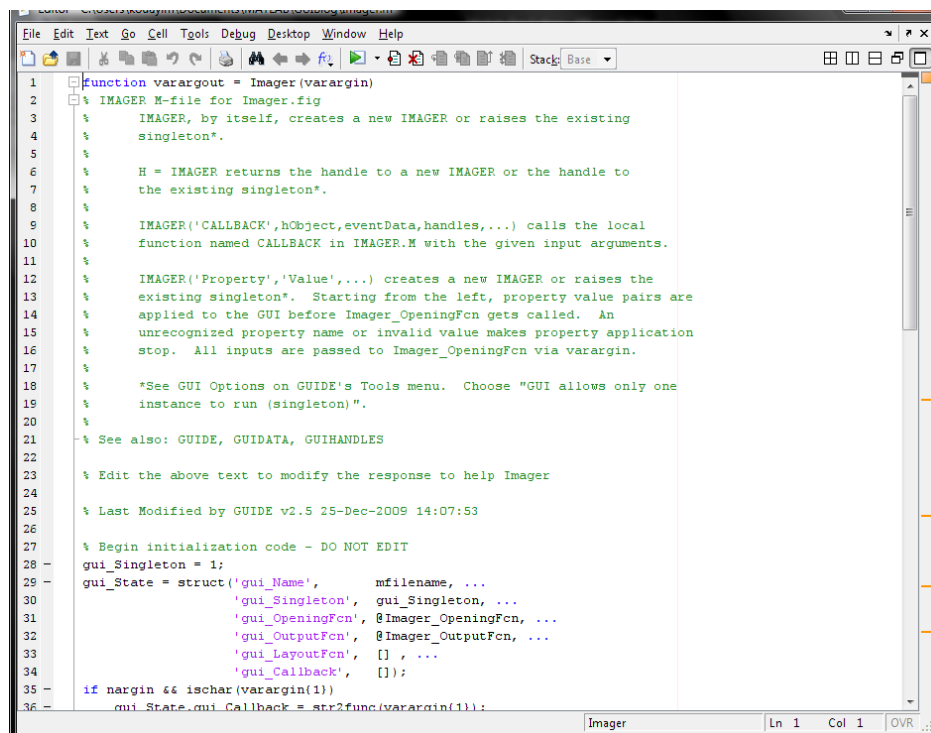
Il existe d'autres possibilités mais nous n'allons pas toutes les citer ici.

Pour notre GUI nous allons simplement mettre un bouton save qui permet d'afficher le chemin (Path) de la photo ouverte dans le fenêtre de commande de Matlab.



Maintenant que nous avons ces trois éléments, nous pouvons commencer à la configurer. Il est conseillé de sauvegarder son travail assez souvent pour éviter de le perdre au cas où Matlab venait à se fermer à cause d'une erreur ou un problème de ce genre. Lors de la première sauvegarde, un nom doit être choisi pour le GUI. Nous prenons simplement « Imager » comme nom.

Quand on sauvegarde notre GUI, il y a un fichier « .m » qui s'ouvre. C'est le programme de notre GUI, et c'est aussi là que nous allons écrire toutes les instructions à exécuter par notre GUI.



### ❖ Configuration de la figure.

Nous devons aussi configurer notre GUI pour qu'il soit plus agréable à voir, pour cela nous allons configurer par exemple la couleur, l'écriture, etc...

Pour accéder à ces paramètres on double-clique deux fois sur l'élément à configurer et une fenêtre s'ouvre. Dans cette fenêtre nous allons changer, par exemple pour notre premier

## Matlab-Exo 1

« Push Button », le paramètre « String » qui est le nom s'affichant sur le bouton. Nous le nommons « Open ». Il est aussi conseillé de changer les « Tag » qui nous permettra d'identifier le bouton lors de la configuration des fonction dans le fichier « .m » de notre GUI. Pour notre bouton « Open » nous prenons « OpenButton » comme Tag. Nous effectuons aussi les changements ci-dessous :

➤ Bouton « Open »

String : **Open**

Tag : **OpenButton**

FontSize : **10.0**

FrontWeight : **bold**

ForegroundColor : **blue**

➤ Bouton « Save »

String : **Save**

Tag : **SaveButton**

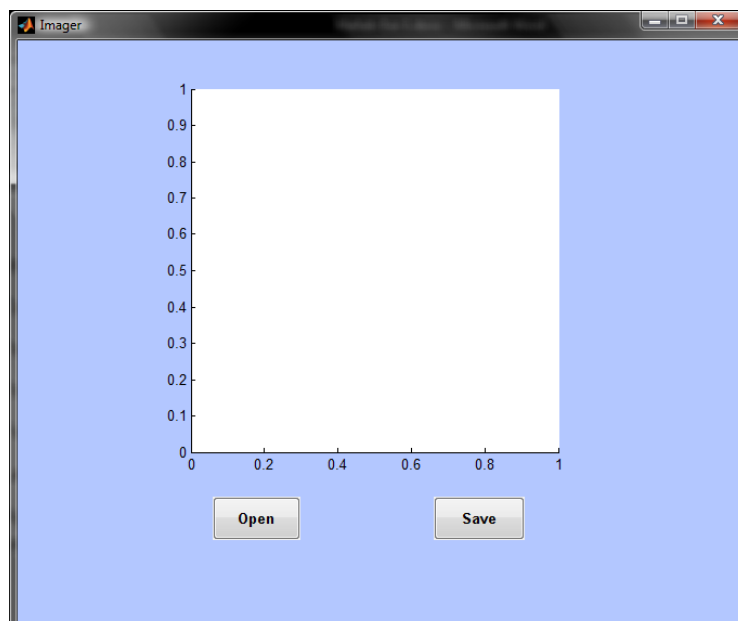
FontSize : **10.0**

FrontWeight : **bold**

ForegroundColor : **blue**

La couleur de fond de notre GUI peut aussi être modifiée.

Après ces modifications, nous pouvons cliquer sur le bouton « run » de notre figure (bouton vert en haut, en forme de petite flèche) et nous obtenons quelque chose qui ressemble à l'image ci-dessous :



Pour le moment notre GUI ne fait toujours rien 😊, mais nous allons maintenant configurer ces fonctions.

## Matlab-Exo 1

### Nous commençons par le bouton « Open » :

Nous cherchons dans le fichier « .m » de notre GUI, la fonction

```
% --- Executes on button press in OpenButton.  
function OpenButton_Callback(hObject, eventdata, handles)  
% hObject    handle to OpenButton (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

Juste après ces instructions on doit entrer toutes les instructions que l'on aimerait voir exécutées lorsque notre bouton « Open » est cliqué. Dans notre cas, nous voulons que une photo soit ouverte. Nous entrons donc l'instruction :

```
[Filename, Pathname] = uigetfile({'*.jpg'; '*.tif'; '*.bmp'; '*.gif'}, 'Select your  
picture(s)', 'MultiSelect', 'off');  
Chemin= Pathname;  
Chemin=strcat(Chemin, Filename);  
Foto=imread(Chemin1); %Lire la Photo à affichée  
h = imshow(Foto);
```

Qui nous permet de sélectionner un fichier Image dans un dossier, d'ouvrir cette photo qui est mise dans la variable « Foto », et qui est ensuite affichées par l'instruction « imshow ».

A ce niveau notre photo est affichée.

### Maintenant passons au bouton « Save » :

Ici nous devons faire des modifications dans plusieurs fonctions.

À la fin des instructions entrées ci-dessus, nous entrons les instructions suivante :

```
LePath = getappdata(0, 'LePath');  
setappdata(LePath, 'PathName', Chemin);
```

La deuxième fonction est celle-ci :

```
% --- Executes just before Imager is made visible.  
function Imager_OpeningFcn(hObject, eventdata, handles, varargin)  
% This function has no output args, see OutputFcn.  
% hObject    handle to figure  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
% varargin   command line arguments to Imager (see VARARGIN)  
  
% Choose default command line output for Imager  
handles.output = hObject;  
  
% Update handles structure  
guidata(hObject, handles);  
  
% UIWAIT makes Imager wait for user response (see UIRESUME)  
% uiwait(handles.figure1);
```

Après ces lignes on ajoute la ligne suivante :

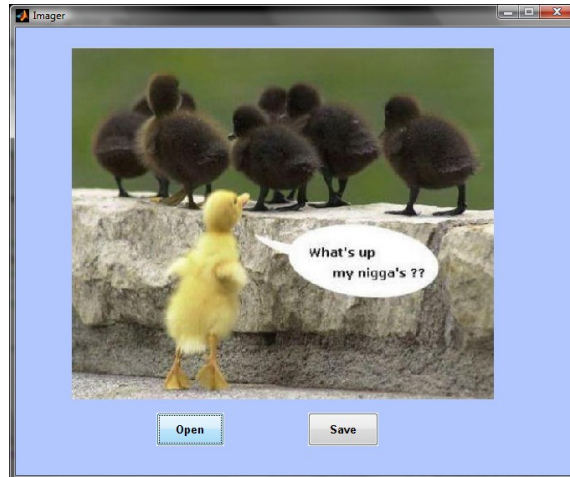
```
setappdata(0 , 'LePath', gcf);
```

## Matlab-Exo 1

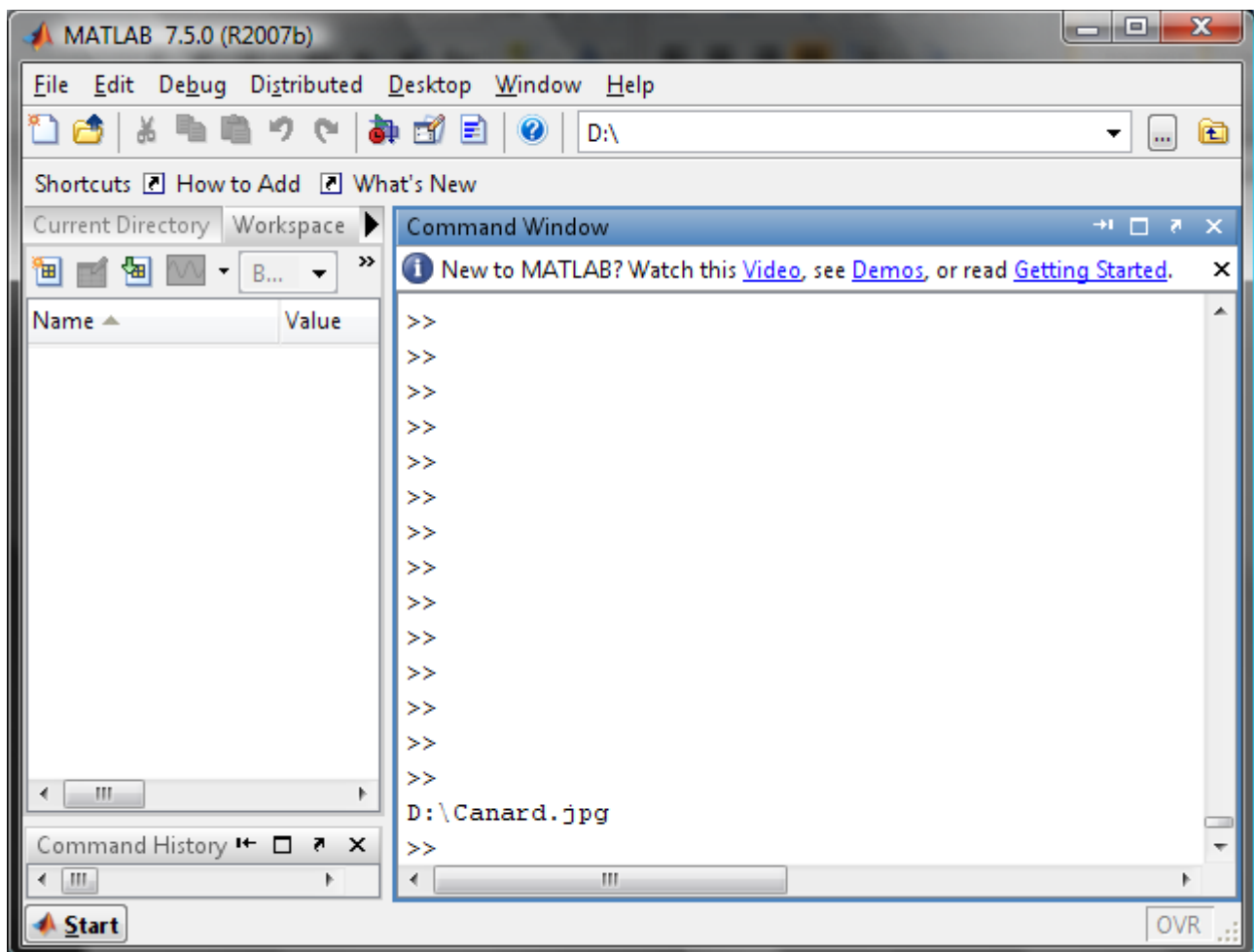
La dernière fonction est celle du bouton « Save ». Elle doit être modifiée comme ci-dessous :

```
% --- Executes on button press in SaveButton.  
function SaveButton_Callback(hObject, eventdata, handles)  
% hObject    handle to SaveButton (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
LePath = getappdata(0, 'LePath');  
PathName = getappdata(LePath, 'PathName');  
disp(PathName);
```

Maintenant notre GUI est terminé. Nous pouvons ouvrir des photos :



En cliquant sur « Save » le chemin vers cette photo est affiché dans la fenêtre de commande de Matlab :



## Matlab-Exo 1

Merci de nous dire si vous avez été satisfait par cette solution pour l'exercice Matlab-Exo 1 soumis par Hadil