

TP N° 9 : interfaces graphiques sur Matlab GUI : Graphical User Interface

Les interfaces graphiques sont des outils puissants qui permettent de créer des applications interactives et conviviales, facilitant ainsi l'interaction entre les utilisateurs et les programmes. Dans ces TP, nous allons voir comment créer une interface graphique en utilisant l'environnement de développement GUIDE (Graphical User Interface Development Environment) de MATLAB.

L'objectif de ce TP est de vous familiariser avec les concepts et les techniques fondamentaux pour créer des interfaces graphiques en MATLAB. Vous apprendrez à concevoir des interfaces graphiques en utilisant des composants tels que les boutons, les zones de texte, les listes déroulantes et les graphiques. Vous découvrirez également comment associer des fonctions de rappel (callbacks) aux événements des composants pour gérer les interactions utilisateur.

Activité 1 : Création d'une application qui calcule la somme de deux nombres

Pour créer une interface graphique avec MATLAB, vous devez utiliser l'outil "GUIDE" (Graphical User Interface Development Environment). Pour ouvrir GUIDE, saisissez dans command window la commande suivante :

```
> guide
```

Dans la fenêtre « GUIDE Quick Start » choisir « Create New GUI » puis « Blanc GUI »

Concevoir l'interface graphique : vous pouvez glisser-déposer les composants nécessaires depuis la palette d'outils sur la fenêtre de la figure. Pour calculer la somme de deux nombres, nous avons besoin d'ajouter deux zones de texte (uicontrol de type "edit"), un bouton (uicontrol de type "pushbutton") et une zone de texte pour afficher le résultat (uicontrol de type "text").

Configurer les propriétés des composants : double-cliquez sur chaque composant et configurez ses propriétés appropriées dans l'onglet "Property Inspector".

Par exemple, vous pouvez attribuer des noms uniques aux zones de texte (par exemple, "editNombre1" et "editNombre2"), au bouton (par exemple, "btnCalculer") et à la zone de texte de résultat (par exemple, "textResultat").

Ajouter le code de calcul de la somme : double-cliquez sur le bouton "btnCalculer" pour ouvrir l'éditeur de code associé à ce bouton. Dans cet éditeur, écrivez le code MATLAB pour effectuer le calcul de la somme en utilisant les valeurs saisies dans les zones de texte. Par exemple :

```
function btnCalculer_Callback(hObject, eventdata, handles)
    nombre1 = str2double(get(handles.editNombre1, 'String'));
    nombre2 = str2double(get(handles.editNombre2, 'String'));
    somme = nombre1 + nombre2;
    set(handles.textResultat, 'String', num2str(somme));
end
```

Exécuter l'application : pour exécuter l'application, cliquer sur le bouton "Exécuter" ou en appuyer sur la touche "F5" de votre clavier. L'application avec l'interface graphique s'ouvrira et vous pourrez saisir les nombres dans les zones de texte, puis cliquer sur le bouton "Calculer" pour afficher la somme dans la zone de texte de résultat.

Explication du code

1. La première ligne définit la fonction **btnCalculer_Callback** qui sera exécutée lorsque le bouton **btnCalculer** est cliqué.
2. Les deux lignes suivantes récupèrent les valeurs saisies dans les zones de texte **editNombre1** et **editNombre2** : La fonction **get(handles.editNombre1, 'String')** récupère la valeur de la zone de texte **editNombre1** et la fonction **str2double** convertit cette valeur en un nombre de type double. **handles** est une structure qui fait référence à l'interface elle-même.
3. La ligne suivante effectue le calcul de la somme en additionnant les deux nombres **nombre1** et **nombre2**. Le résultat est stocké dans la variable **somme**.
4. Enfin, la dernière ligne utilise la fonction **set** pour mettre à jour le contenu de la zone de texte **textResultat** avec la valeur de la somme convertie en une chaîne de caractères à l'aide de la fonction **num2str(somme)**.

Activité 2 : Création d'une application de conversion de température entre Celsius et Fahrenheit

Voici un autre exemple d'application avec une interface graphique qui effectue une conversion de température entre Celsius et Fahrenheit.

1. Création de l'interface graphique : Créer une « Blanc GUI » en suivant les mêmes étapes de l'activité 1.
2. Concevoir l'interface graphique : ajoutez deux zones de texte (uicontrol de type "edit") pour saisir la température, deux boutons radio (uicontrol de type "radiobutton") pour sélectionner l'unité de température (Celsius ou Fahrenheit) et un bouton (uicontrol de type "pushbutton") pour effectuer la conversion. Ajoutez également une zone de texte (uicontrol de type "text") pour afficher le résultat.
3. Configurer les propriétés des composants : sélectionnez chaque composant et configurez ses propriétés appropriées dans l'onglet "Property Inspector". Attribuez des noms uniques aux zones de texte "editTemperature", aux boutons radio "radioCelsius" et "radioFahrenheit", au bouton de conversion "btnConvertir" et à la zone de texte de résultat "textResultat".
4. Ajouter le code de conversion de température : double-cliquez sur le bouton "btnConvertir" pour ouvrir l'éditeur de code associé à ce bouton. Dans cet éditeur, écrivez le code MATLAB pour effectuer la conversion de température en fonction des valeurs saisies et de l'unité sélectionnée :

```
function btnConvertir_Callback(hObject, eventdata, handles)
    temperature = str2double(get(handles.editTemperature, 'String'));

    if get(handles.radioCelsius, 'Value')
        fahrenheit = (temperature * 9/5) + 32;
        set(handles.textResultat, 'String', num2str(fahrenheit) + ' °F');
    elseif get(handles.radioFahrenheit, 'Value')
        celsius = (temperature - 32) * 5/9;
        set(handles.textResultat, 'String', num2str(celsius) + ' °C');
    end
end
```

Ensuite exécuter l'application

Activité 3 : Boite de dialogue

Les boîtes de dialogue sont des fenêtres modales qui s'affichent à l'écran pour demander à l'utilisateur d'entrer des informations, de prendre une décision ou de confirmer une action. MATLAB propose différentes fonctions pour créer des boîtes de dialogue adaptées à diverses situations. Voici une description des principales boîtes de dialogue disponibles :

1. **inputdlg** : Cette fonction permet de créer une boîte de dialogue qui demande à l'utilisateur d'entrer une ou plusieurs valeurs. Elle peut être utilisée pour récupérer des données saisies par l'utilisateur, telles que des nombres, des chaînes de caractères, etc. La fonction `inputdlg` retourne les valeurs saisies par l'utilisateur sous forme de cellule.

Exemple :

Exécuter le code suivant :

```
prompt = 'Entrez un nombre : ';\ndlgtitle = 'Bo?te de dialogue';\ndims = [1 35];\ndefinput = {'0'};\nresult = inputdlg(prompt, dlgtitle, dims, definput);\ndisp(['La valeur entr?e est : ', result{1}]);
```

Explication :

- La variable **prompt** contient le message qui sera affiché dans la boîte de dialogue pour demander à l'utilisateur d'entrer une valeur.
 - La variable **dlgtitle** représente le titre de la boîte de dialogue.
 - La variable **dims** définit les dimensions de la boîte de dialogue (nombre de lignes et de colonnes).
 - La variable **definput** est une cellule contenant la valeur par défaut à afficher dans la boîte de dialogue.
 - La fonction **inputdlg** est utilisée pour créer la boîte de dialogue en prenant en entrée les différentes variables précédemment définies. Le résultat est stocké dans la variable **result**.
 - Afficher le résultat : Vous pouvez utiliser la fonction **disp** pour afficher la valeur entrée par l'utilisateur dans la boîte de dialogue.
2. **msgbox** : Cette fonction permet de créer une boîte de dialogue d'information qui affiche un message à l'utilisateur. Elle est généralement utilisée pour afficher des messages, des avertissements ou des informations à l'utilisateur. La boîte de dialogue peut contenir un titre optionnel et des icônes pour indiquer le type d'information.

Exemple :

Exécuter le code suivant :

```
msg = 'Bonjour, bienvenue dans notre programme !';\ntitre = 'Message de bienvenue';\nmsgbox(msg, titre);
```

Explication :

Les paramètres de la fonction `msgbox` sont les suivants :

- **msg** : La chaîne de caractères contenant le message que vous souhaitez afficher à l'utilisateur.
- **titre** : Le titre de la boîte de dialogue.

Lorsque vous exécutez ce code, une boîte de dialogue s'affiche avec le message "Bonjour, bienvenue dans notre programme !" et le titre "Message de bienvenue". L'utilisateur peut cliquer sur le bouton "OK" pour fermer la boîte de dialogue.

3. **warndlg** : Cette fonction crée une boîte de dialogue d'avertissement qui informe l'utilisateur d'une situation ou d'une condition particulière. Elle est souvent utilisée pour signaler des erreurs, des avertissements ou des problèmes potentiels. Comme msgbox, elle peut également inclure un titre et des icônes pour renforcer le message.

Exemple :

Exécuter le code suivant :

```
msg = 'Attention ! La valeur saisie est incorrecte.';
titre = 'Avertissement';
warndlg(msg, titre);
```

4. **errordlg** : Cette fonction crée une boîte de dialogue d'erreur pour signaler des erreurs ou des problèmes importants à l'utilisateur. Elle est utilisée lorsque quelque chose d'inattendu se produit ou lorsque les conditions requises ne sont pas remplies. Elle peut être accompagnée d'un titre et d'icônes pour indiquer la gravité de l'erreur.

Exemple :

Exécuter le code suivant :

```
msg = 'Une erreur s est produite. Veuillez réessayer.';
titre = 'Erreur';
errordlg(msg, titre);
```

5. **questdlg** : Cette fonction permet de créer une boîte de dialogue de question qui demande à l'utilisateur de choisir parmi plusieurs options. Elle est utilisée lorsque vous avez besoin d'une réponse spécifique de la part de l'utilisateur, telle qu'une confirmation ou un choix parmi plusieurs alternatives. La fonction questdlg retourne l'option sélectionnée par l'utilisateur.

Exemple :

Exécuter le code suivant :

```
question = 'Voulez-vous enregistrer les modifications ?';
titre = 'Enregistrer les modifications';
options = {'Oui', 'Non', 'Annuler'};
selectedOption = questdlg(question, titre, options);
```

Activité 4 : Création d'un menu

Ce code permet de créer une fenêtre de figure avec une barre de menu contenant un menu "option" qui contient trois options de menu (Option 1, Option 2, Option 3).

```
% Creation d'une figure
fig = figure('Name', 'Menu Example', 'MenuBar', 'none', 'Position', [100
100 300 200]);

% Creation d'un menu
menuBar = uimenu(fig, 'Label', 'Options');
% Ajout des elements de menu
menu1 = uimenu(menuBar, 'Label', 'Option 1', 'Callback', @option1Callback);
menu2 = uimenu(menuBar, 'Label', 'Option 2', 'Callback', @option2Callback);
menu3 = uimenu(menuBar, 'Label', 'Option 3', 'Callback', @option3Callback);
```

Explication :

Nous créons une fenêtre de figure en utilisant la fonction **figure**. Ensuite, nous créons une barre de menu en utilisant la fonction **uimenu**, en spécifiant le nom du menu principal avec le paramètre '**Label**'. Ensuite, nous ajoutons des éléments de menu à la barre de menu en utilisant à nouveau la fonction **uimenu**. Nous spécifions le nom de chaque option avec le paramètre '**Label**' et définissons une fonction de rappel (callback) pour chaque option en utilisant le paramètre '**Callback**'. Les fonctions de rappel sont appelées lorsque l'utilisateur sélectionne une option de menu spécifique.

Lorsque l'utilisateur sélectionne l'une des options de menu, un message d'erreur sera afficher

```
Undefined function 'option2Callback' for input  
arguments of type 'double'.
```

```
Error while evaluating uimenu Callback
```

Cela parce que l'option2 fait appel à la fonction **option2Callback** qui n'est pas encore définie. Il faut alors définir les trois fonctions : **option1Callback**, **option2Callback**, **option3Callback**, que vous avez appelé dans le code de création de menu dans des fichiers séparés.

Exemple : la fonction **option1Callback** permet d'afficher un message dans command Window

```
function option1Callback(~, ~)  
    disp('Option 1 selectionnee');  
end
```

maintenant vous pouvez tester l'option 1 du menu