

Comment créer votre première interface avec Tk

par Rodrigue Hunel ([Site perso](#))

Date de publication : 21/01/2008

Il existe de nos jours plusieurs bibliothèques permettant de générer des interfaces pour Perl. La plus simple à utiliser est Tk. Ce tutoriel aura donc pour but de vous aider à créer votre première interface avec cette bibliothèque.

I - Introduction

I-A - Présentation de Tk

I-B - Installation de la bibliothèque

II - L'interface

II-A - Création de la fenêtre

II-B - Création d'une barre de menus

II-C - Création du texte "Hello World"

II-D - Création d'une barre de statut

II-E - Gestion des événements

II-F - Code final

III - Conclusion



III-A - Remerciements

III-B - Liens

I - Introduction

I-A - Présentation de Tk

Tk est une bibliothèque graphique multi-plateforme créée, à l'origine, pour la création d'interface pour les scripts en langage Tcl et est de plus en plus utilisée avec divers langages dont notre langage préféré Perl.


 Pour plus d'informations, n'hésitez pas à visiter le site s'y rapportant :  **Tcl/Tk**

I-B - Installation de la bibliothèque

Cette bibliothèque étant accessible depuis n'importe quel système d'exploitation, nous pouvons donc l'installer sous Windows tout comme sous Linux.

- Sous Windows, si vous disposez déjà du package **ActivePerl** distribué par **ActiveState**, aucune installation n'est nécessaire. Sinon, je vous conseille de l'installer afin de bénéficier de Tk.
- Sous Linux et à l'aide du terminal, saisissez la commande qui suit pour lancer l'installation.

```
sudo apt-get install perl-tk
```


 On peut également le faire à partir du "Gestionnaire de paquets" en effectuant une recherche avec les mots-clés **perl** et **tk**.

II - L'interface

Nous allons, enfin, commencer à créer notre interface, mais avant tout codage il serait bon de connaître notre cahier des charges.

Notre but sera d'obtenir une simple fenêtre contenant :

- Une barre de menu ;
- Un label "Hello World" ;
- Une barre de statut contenant "Hello World" ;
- Et une simple gestion d'évènements.

 *Avant de commencer, je tiens à préciser que ce développement a été réalisé sous Windows mais est entièrement compatible Linux/Unix.*

II-A - Création de la fenêtre

Avant tout code, il ne faut pas oublier de spécifier que l'on utilise la bibliothèque Tk, en ajoutant la ligne suivante.

Code Perl/Tk

```
use Tk; #Utilisation de Tk
```

Pour créer notre fenêtre, nous devons faire appel au Widget **MainWindow**.

Code Perl/Tk

```
$fen = MainWindow->new(); #Création de la fenêtre principale
```

Ce qui a pour effet de créer une nouvelle instance (référence, pointeur) vers notre nouvelle fenêtre.

Afin de la personnaliser, nous pouvons lui spécifier un titre.

Code Perl/Tk

```
$fen->title("Interface Tk avec Perl"); #Définition du titre de la fenêtre
```

Ainsi, notre code sera :

Code Perl/Tk

```
#!/usr/local/bin/perl
use Tk; #Utilisation de Tk

$fen = MainWindow->new(); #Création de la fenêtre principale
$fen->title("Interface Tk avec Perl"); #Définition du titre de la fenêtre

MainLoop;
```

i La fonction `MainLoop` correspond à l'appel du gestionnaire d'évènements. Cette dernière ligne est indispensable car c'est elle qui achève, obligatoirement, notre script Tk. De plus son absence empêcherait l'affichage correct de notre interface.



Une fenêtre vide

II-B - Création d'une barre de menus

Maintenant que l'on a notre fenêtre, on va lui rajouter un menu contenant un bouton "Quitter".

Pour ce faire, nous allons utiliser les Widgets :

- **Frame** qui créera notre barre ;
- **Menubutton** qui créera notre menu.

On rajoute donc le code suivant :

Code Perl/Tk

```
#Création de la barre de menu
$barre_menu = $fen->Frame(-relief => 'groove' , -borderwidth => 2);
```

Nous obtiendrons une barre vide avec une bordure de 2px et un effet de relief pour bien différencier notre élément et le reste de la fenêtre.

Ensuite, il ne nous reste plus qu'à rajouter notre bouton.

Code Perl/Tk


```
#Création de la barre du menu général
$menu_app = $barre_menu->Menubutton(
    -text => 'Application', #Définition du libellé du bouton
    -font => '{Garamond} 10', #Définition de la police
    -tearoff => 0 ,
    -menuitems => [ #Ajout des items à notre menu
        [ 'command' => 'Quitter' ,
          -font => '{Garamond} 10',
        ]
    ]
);
```

En laissant le code tel quel, nous n'aurons aucun effet à l'affichage. Pour dire à notre application de prendre en compte la création de nos Widget, il faut rajouter les deux lignes :

Code Perl/Tk

```
#Affichage du menu général à gauche
$menu_app->pack(-side => 'left');

#Affichage de la barre de menu ancrée en haut et sur toute la largeur de la fenêtre
$barre_menu->pack(-side => 'top', -anchor => 'n', -fill => 'x');
```

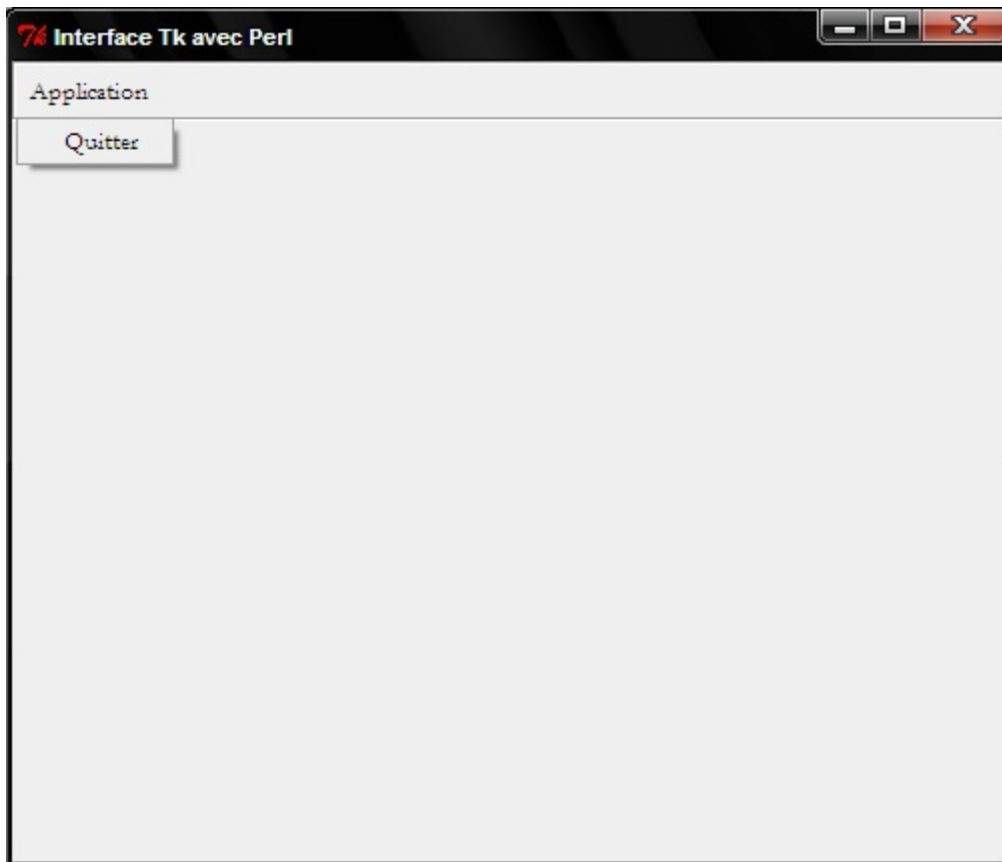
 *Sous Tk, il existe quatre gestionnaires de positionnement (autrement appelés gestionnaires de géométrie) de Widgets :*

- form ;
- grid ;
- place ;
- pack.

Ici, nous utilisons le plus facile à mettre en oeuvre, le packer(pack) dont le principe de fonctionnement est basé sur l'ordre de déclaration des Widgets. Autrement dit, tous nos widgets seront positionnés à la suite tout en respectant les options qui lui sont définies.

- **-side**, permet de définir la position à laquelle notre Widget sera placé. Soit à gauche (left), à droite (right), en haut (top, qui est la valeur par défaut) et en bas (bottom) ;
- **-fill**, précise le type de remplissage adopté par notre élément. Il peut être sur la largeur (x), la hauteur (y), les deux (both) et aucun des deux (none) ;
- **-anchor**, règle son point d'ancrage dans la fenêtre. Ses valeurs sont n (nord), s (sud), e (est), w (ouest), center (centrer), sw (sud-ouest), se (sud-est), ne (nord-est), nw (nord-west) ;
- **-expand** qui permet d'attribuer tout l'espace restant au dernier widget placé (que ce soit en largeur ou en hauteur). Valeurs possibles : x ou y.

Voici l'interface obtenue :



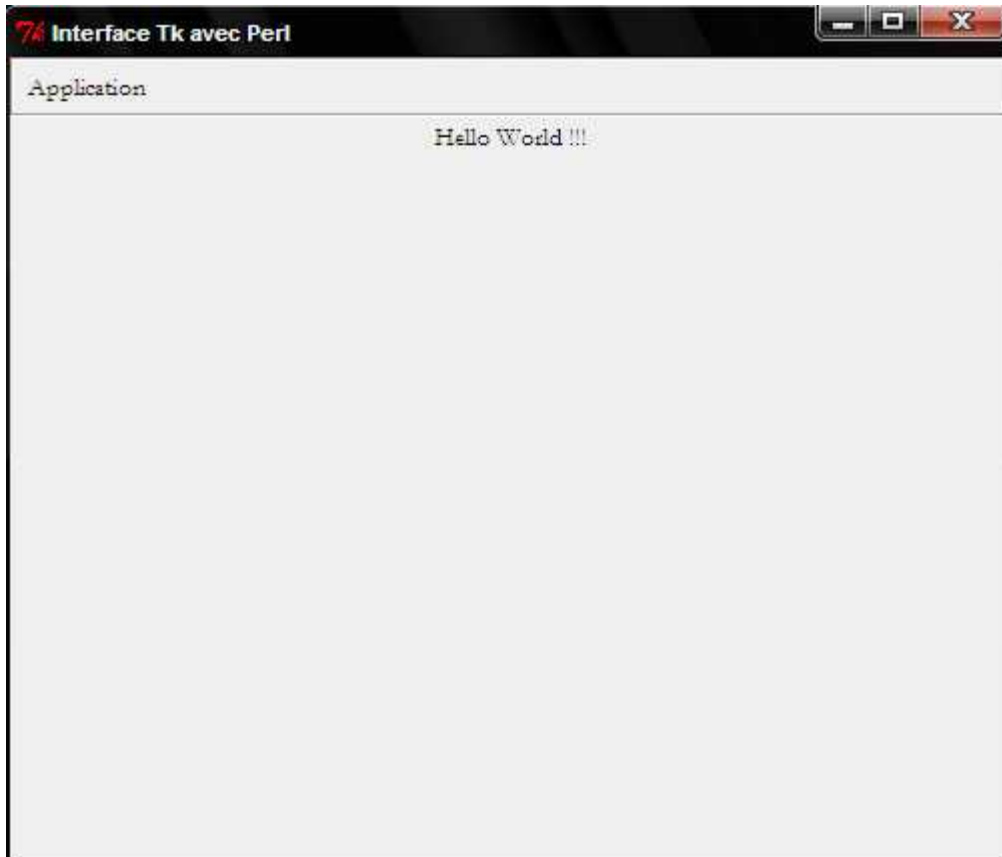
Une fenêtre avec une barre de menu

II-C - Création du texte "Hello World"

Tout comme les autres, pour ajouter du texte il faut utiliser un Widget. Dans notre cas ce sera **Label**, qui ajoute un texte statique.

Code Perl/Tk

```
#Ajout du label Hello World
$fen->Label(
    -text => "Hello World !!!" ,
    -font => '{Garamond} 10' ,
)->pack(
    -anchor => 'ne' ,
    -fill => 'x'
);
```



Une fenêtre avec du texte

II-D - Création d'une barre de statut

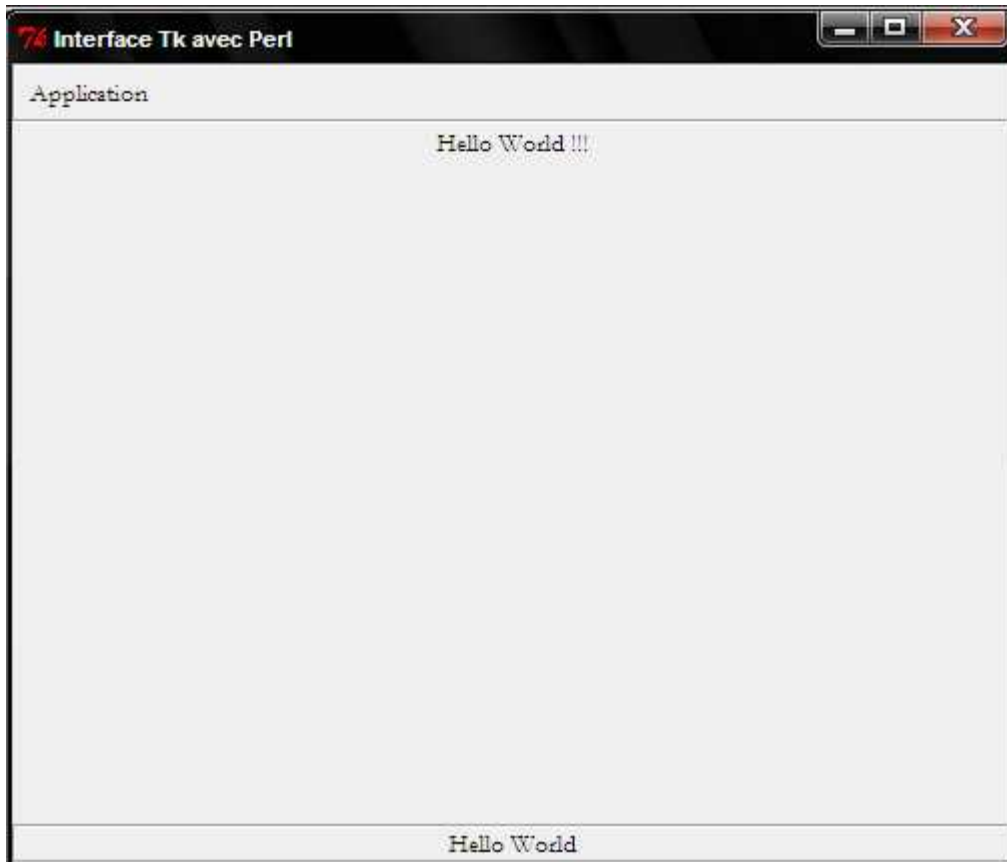
Sous Tk, il n'existe malheureusement pas de Widget prédéfini pour en créer. Par conséquent, nous devons *jouer* avec un autre : le Widget **Label**.

Vu que l'on utilise **Label**, le code de création sera identique tandis que celui de l'ajout, non.

Code Perl/Tk

```
#Création de notre barre de statut à l'aide d'un Label
$barre_statut = $fen->Label(-relief => 'groove', -text => "Hello World", -font => '{Garamond} 10');

#On l'ajoute en bas de la fenêtre afin de lui donner l'effet barre de statut
$barre_statut->pack( -side => 'bottom', -fill => 'both');
```



Une fenêtre avec une barre de statut

II-E - Gestion des évènements

Un évènement est produit lors d'un clic sur un bouton ou toute autre action produite ; par l'utilisateur.

Ici, notre action sera de demander à notre bouton *Quitter* de fermer notre fenêtre.

Il suffit dans ce cas de modifier le code précédent du bouton, en rajoutant l'option **-command**. En effet, c'est elle qui lie l'action à effectuer à notre bouton.

Il existe deux utilisations possibles avec cette option :

- La première consiste à définir, avec la méthode inline, les actions à effectuer ;

Utilisation de -command avec la méthode inline

```
-command => sub { exit(0); } #Cette définition appellera directement la fonction exit prédéfinie  
chez Tk
```

- La seconde permet de faire appel à des fonctions externes à la définition de notre Widget.

Définition de notre fonction de fermeture de l'application

```
#Définition de notre fonction de fermeture de l'application
```

Définition de notre fonction de fermeture de l'application

```
sub Quitter
{
    exit(0);
}
```

Utilisation de -command avec une référence sur notre fonction

```
-command => \&Quitte #Cette définition appellera notre fonction Quitte
```

Code résultant :

Code Perl/Tk

```
#Création de la barre du menu général
$menu_gen = $barre_menu->Menubutton(
    -text => 'Application' ,
    -font => '{Garamond} 10' ,
    -tearoff => 0 ,
    -menuitems => [
        [ 'command' => 'Quitte' ,
          -font => '{Garamond} 10' ,
          -command => \&Quitte
        ]
    ]
);
```

II-F - Code final

Le code correspondant à notre interface terminée est

Code Perl/Tk

```
#!/usr/local/bin/perl
use Tk; #Utilisation de Tk

#Définition de notre fonction de fermeture de l'application
sub Quitte
{
    exit(0);
}

#Création de la fenêtre principale
$fen = MainWindow->new();

#Définition de la taille min de la fenêtre
$fen->minsize('500','400');

#Définition du titre de la fenêtre
$fen->title("Interface Tk avec Perl");

#Création de la barre de menu
$barre_menu = $fen->Frame(-relief => 'groove' , -borderwidth => 2);

#Création de la barre du menu général
$menu_gen = $barre_menu->Menubutton(
    -text => 'Application' ,
    -font => '{Garamond} 10' ,
    -tearoff => 0 ,
```

Code Perl/Tk

```
-menuitems => [
    [ 'command' => 'Quitter' ,
      -font => '{Garamond} 10',
      -command => \&Quitter
    ]
];

#Affichage du menu général à gauche
$menu_gen->pack(-side => 'left');

#Affichage de la barre de menu en haut de la fenêtre
$barre_menu->pack(-side => 'top', -anchor => 'n', -fill => 'x');

#Ajout du label Hello World
$fen->Label(
    -text => "Hello World !!!" ,
    -font => '{Garamond} 10',
)->pack(
    -anchor => 'ne',
    -fill => 'x'
);

#Création de notre barre de statut à l'aide d'un Label
$barre_statut = $fen->Label(-relief => 'groove', -text => "Hello World", -font => '{Garamond} 10');

#On l'ajoute en bas de la fenêtre afin de lui donner l'effet barre de statut
$barre_statut->pack( -side => 'bottom', -fill => 'both');

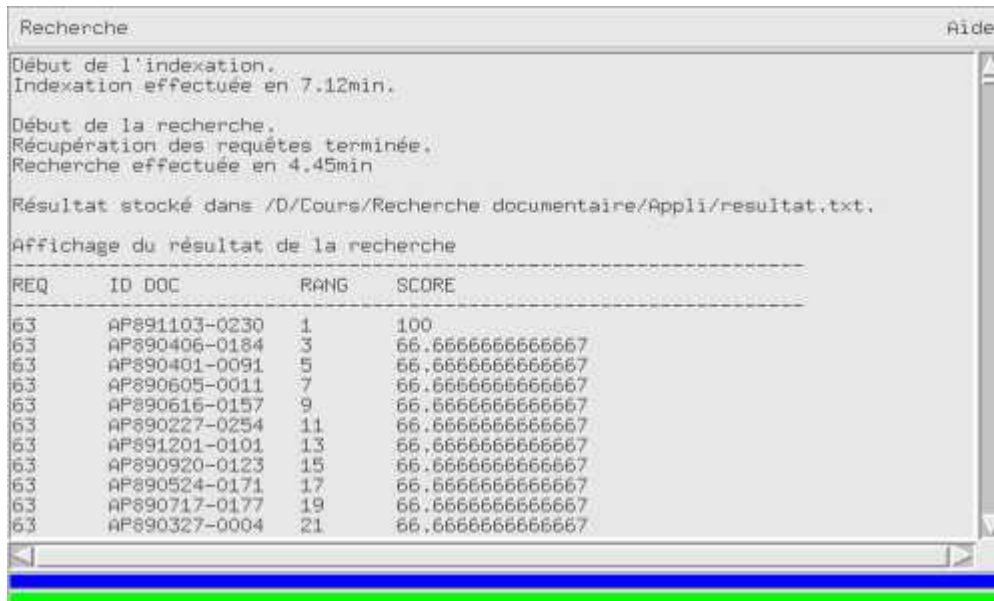
MainLoop;
```



III - Conclusion

Voilà, vous avez maintenant réalisé votre première interface graphique avec Tk. Comme vous avez pu le remarquer, ce fut assez facile, et ce, grâce à la simplicité d'utilisation des Widgets qu'offre cette bibliothèque.

Bien sûr, Tk ne se limite pas qu'à ce genre d'interface. On peut aller beaucoup plus loin en rajoutant divers Widgets. Je vous propose un exemple, en deux images, que j'ai tiré d'une de mes dernières réalisations.



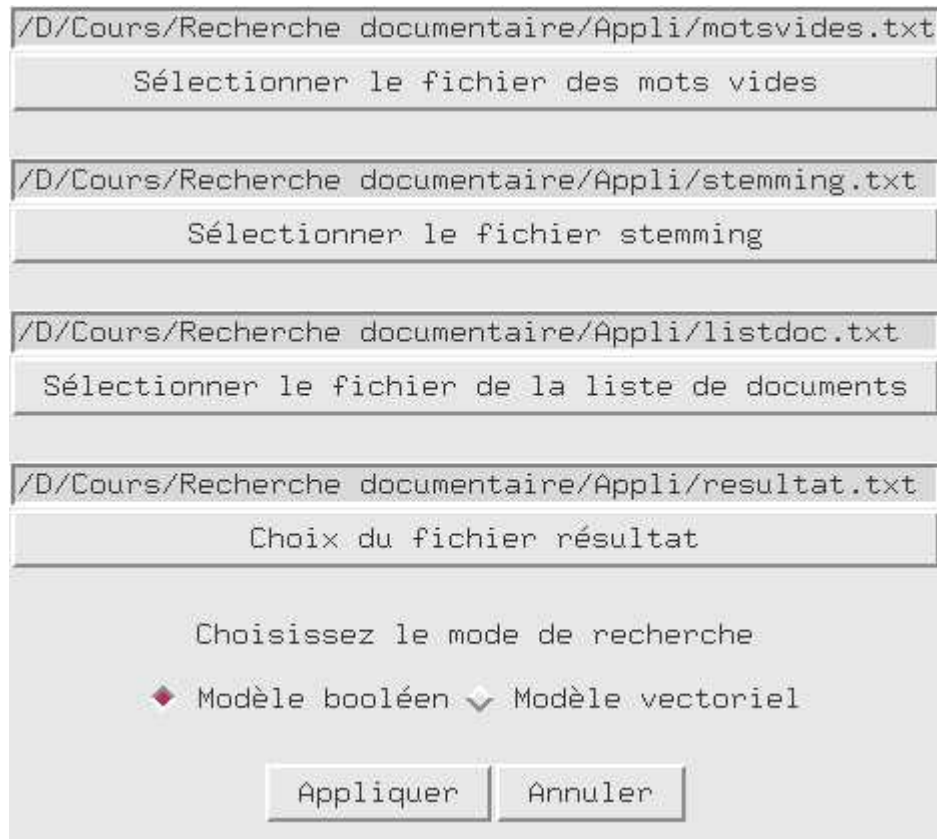
```
Recherche. Aide
Début de l'indexation.
Indexation effectuée en 7.12min.

Début de la recherche.
Récupération des requêtes terminée.
Recherche effectuée en 4.45min

Résultat stocké dans /D/Cours/Recherche documentaire/Appli/resultat.txt.

Affichage du résultat de la recherche
-----
REQ      ID DOC      RANG      SCORE
-----
63      AP891103-0230  1         100
63      AP890406-0184  3         66.6666666666667
63      AP890401-0091  5         66.6666666666667
63      AP890605-0011  7         66.6666666666667
63      AP890616-0157  9         66.6666666666667
63      AP890227-0254  11        66.6666666666667
63      AP891201-0101  13        66.6666666666667
63      AP890920-0123  15        66.6666666666667
63      AP890524-0171  17        66.6666666666667
63      AP890717-0177  19        66.6666666666667
63      AP890327-0004  21        66.6666666666667
```

Interface avec menu, zone de texte, et barres de progression



Interface d'options liées à un fichier ini

III-A - Remerciements

Tous mes remerciements à **LineLe** et à **Woufeil** pour leur relecture.

III-B - Liens

 [Tous les cours Perl de Developpez.com](#)

 [La documentation Tcl/Tk](#)

 [Le wiki francophone sur Tcl/Tk](#)

