



TD Interface graphique

Université de Provence

Licence Mathématiques Informatique- PEIP

Année 2011-2012

Le module Tkinter donne des fonctionnalités permettant de créer des interfaces graphiques.

Exercice 1 Fenêtre Labels et Buttons.

La fonction `.Button()` renvoie une référence vers un bouton qui appelle la fonction indiquée dans l'argument `command=`.

1. Ecrire le script qui permet de créer une fenêtre de taille 400x60, appelée "*Des chiffres et des lettres*" et affiche *le compte est bon* dans un champ texte. Ensuite le script affiche tapez "*entrée*" pour terminer.
2. Modifier le script pour qu'il affiche tapez "*entrée*" pour avancer, modifier l'affichage en *le mot le plus long*. Ensuite le script affiche tapez "*entrée*" pour terminer.
3. Modifier le script précédent pour qu'il affiche un bouton *épreuve suivante* qui transforme un affichage "*le compte est bon*" en un affichage "*le mot le plus long*" et réciproquement. Le script affiche tapez "*entrée*" pour terminer.

Exercice 2 La fonction `.get()` appelée sur la référence d'un widget `Entry` retourne la chaîne de caractères contenue dans celui-ci.

1. Créer une fenêtre avec un titre *exercice 2*, une taille fixe, et un Label *label*, un champ de saisie (widget `Entry`) *entree* et un Button *bouton*.
2. Ecrire une fonction *modifie_label* qui transforme le texte du Label *label* en la valeur d'un compteur *cpt* (qui sera une variable globale) et s'incrémente à chaque utilisation du bouton.
3. Modifier la fonction appelée par le bouton pour lui faire écrire le contenu du champ de texte dans le label.

Exercice 3 Codage des couleurs : une couleur est codée par un nombre en hexadécimal comme dans `coul="#0e0"` ou directement par son nom (pour certaines couleurs prédéfinies) comme dans `coul="red"`.

1. Créez une fenêtre *fen* de taille fixe. Créez une liste *listeLabels* de *n* éléments contenant des références vers des widgets `Label`. Tous les éléments de la liste auront les mêmes dimensions mais des couleurs différentes.
2. Utilisez `.grid(column=x, row=y)` pour placer ces éléments dans la fenêtre (*x* et *y* représentent des nombres entiers). La liste sera affichée sur une seule ligne.
3. Créer un bouton, le placer en fin de ligne à l'aide de `.grid()`.

4. Définir une fonction *tourner()* qui décale d'une case vers la droite l'affichage de chaque élément de la liste (l'élément affiché le plus à droite sera affiché tout à gauche). Cette fonction sera appelée à chaque clic sur le bouton.

Remarque : les appels successifs à *.grid()* (ou *.pack()*) avec des paramètres différents ne permettent pas de déplacer l'affichage de l'élément (ne duplique pas l'affichage). Mettre à jour l'affichage des éléments : un appel à *.grid()* ou *.pack()* devrait toujours (ou presque) être suivi d'un appel à la fonction *.update()*, ici sur la fenêtre principale : *fen.update()*

Attention : l'utilisation de *.grid()* et de *.pack()* pour placer différents éléments dans une même zone d'affichage (ici la fenêtre) est susceptible de lancer Tk dans une boucle infinie, les mélanges sont donc fortement déconseillés.

Exercice 4 La fonction *.bind()* permet d'associer la réception d'un évènement en provenance du clavier ou de la souris avec un appel de fonction, comme dans *fen.bind("jButtonPress", modifie)* qui associe un clic de souris à la fonction *modifie*. La liste des chaînes de caractères acceptées en premier paramètre peut être obtenue dans l'aide : *help("Tkinter.Tk.bind")*. La fonction appelée ne peut contenir qu'un seul argument qui est la référence vers l'évènement à l'origine de l'appel comme *def modifie(Evt) : ...*, voir *help("Tkinter.Event")*. Si *Evt* est un évènement alors *Evt.x* et *Evt.y* sont les coordonnées correspondant à l'évènement.

1. Ecrire une fonction *modifie(Evt)* qui modifie le texte d'un Label *label* en le remplaçant par le message "clic de souris reçu en x et y" où x et y sont les coordonnées du point sur lequel on a cliqué.
2. Ecrire un script qui crée une fenêtre avec un Label qui contient le text 'rien reçu', puis affiche les coordonnées d'un clic de souris effectué dans la fenêtre.

Exercice 5 Un début de jeu de serpent. Un serpent est positionné sur une grille et se dirige vers une proie qui est sur la grille. Le serpent se déplace selon les 4 directions *haut*, *bas*, *gauche*, *droite*.

1. Utiliser une liste de listes de labels positionnés en grille pour représenter la surface de jeu dans une fenêtre Tk().
2. Ajouter un nouveau label représentant le serpent et l'affiche par dessus le label qui est au centre de la grille.
3. Configurer la réception de 4 touches de claviers (au choix) pour déplacer le serpent vers le haut, le bas, la gauche ou la droite. Que se passe-t-il lorsque le serpent quitte le plateau de jeu ?
4. Placer aléatoirement sur la grille un label qui sera la cible du serpent. Le programme se termine lorsque le serpent atteint la cible.