LMI 1 – Semestre 2

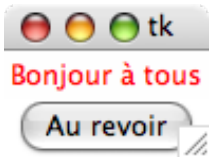
Université d'Artois
Faculté des Sciences Jean Perrin

ALGO2 – Algorithmique et Programmation 2

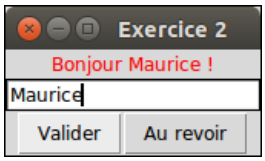
Fiche TP numéro 5

Pour commencer

Exercice 1 : Écrire une classe `VueMessage` qui créé une fenêtre graphique avec un message de bienvenue et un bouton pour quitter l'application. Votre application doit donc contenir une étiquette (`Label`), et un bouton. Un message de bienvenue est proposé par défaut, mais il peut être donné en paramètre du constructeur de `VueMessage`.



Exercice 2 : On complique l'application précédente : on veut maintenant dire bonjour à la personne qui aura donné son nom.



1. Créez la classe `VueMessage2` qui construit une fenêtre avec chacun des composants graphiques élémentaires nécessaires : une étiquette, une zone de saisie (`Entry`), deux boutons
2. Améliorez l'organisation du placement pour que les deux boutons soient côte à côte.
3. Spécifiez puis écrivez la méthode `valider` qui change le message de bienvenue en fonction du nom saisi par l'utilisateur. Quels sont les composants auxquels vous avez besoin d'accéder dans cette méthode ? Modifiez votre constructeur afin que ces composants soient mémorisés chacun dans un attribut.
4. Associez la méthode `valider` à un clic sur le bouton `valider`.
5. N'oubliez pas de lancer la boucle d'écoute des événements à la fin de votre constructeur.

On s'approche du démineur

Exercice 3 :

— Téléchargez à partir de Moodle les images nécessaires pour ce TP.
— Toutes ces images ont une dimension de 20 pixels sur 20 pixels. Déclarez une constante `DIM` qui vaut 21 (on prend un peu plus de place pour anticiper un peu les exercices suivants).

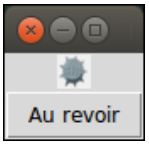
— Créez la classe `VueDemineur` qui construit une fenêtre avec deux boutons : le premier affiche `demineur/mine.gif`, le second est un bouton *Au revoir* qui permet de quitter l'application. Pour l'image, vous n'oublierez pas, après avoir créé votre fenêtre de type `Tk()`, de créér une instance de `PhotoImage` à partir du fichier `mine.gif`. L'élément principal va maintenant être un objet canevas `Canvas` :

```
can = tkinter.Canvas(fenetre,width=DIM,height=DIM)
```

Puis vous allez poser l'instance de `PhotoImage` sur le canevas, en collant son coin haut gauche (nord-ouest) aux coordonnées `(0,0)` :

```
can.create_image(0,0,anchor=tkinter.NW,image = image_mine)
```

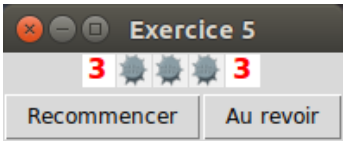
N'oubliez pas de lancer la boucle d'écoute des événements à la fin de votre constructeur. Écrivez le script principal qui lance votre application.



— On veut maintenant afficher une image du démineur au hasard.
Vous devez stocker les images du démineur (sous la forme de `PhotoImage`) dans une liste, dans l'ordre suivant : rien, un, deux, trois, . . ., huit, mine. L'image `cache.gif` n'est pas concernée pour le moment. Vous devrez spécifier et écrire une fonction `initialisationImages` qui effectue cette initialisation. Cette fonction sera appelée dans le constructeur de la classe `VueDemineur`. **Attention :** cette fonction sera appelée après avoir créé une fenêtre `Tk()`. Complétez votre code pour afficher une image en tirant aléatoirement une valeur dans l'intervalle [-1 ;8].

Exercice 4 : On veut maintenant afficher cinq images, tirées aléatoirement parmi les images du démineur (en dehors de `cache.gif`). Elles sont affichées côte à côte sur le même canevas, dont la largeur doit être maintenant de $5 \times$`DIM`.

Exercice 5 : On reprend l'exercice précédent mais on peut rejouer et ré-afficher aléatoirement les images.



Exercice 6 : Reprenez l'exercice précédent, mais avec un affichage des images qui se fait sur 5 lignes et 5 colonnes.