UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Outils Info. TP Chomp

Rendu : sous forme d'archive

À la fin, vos réponses seront rendues dans une archive zip : cette archive contiendra un fichier `compte-rendu.txt` (avec votre **nom**, **groupe**, et les réponses aux questions précédées d'une apostrophe) et vos fichiers pythons.

Pour faire une **archive zip** de votre TP pour le rendu, exécutez, dans le terminal :

```
cd ~/outils-info/
zip -r tpchomp.zip tpchomp/
```

Important : mise en place

- travaillez dans un dossier dédié au TP, lui même dans `~/outils-info`,
- créez un fichier `compte-rendu.txt` pour écrire la date, vos noms et les réponses aux questions précédées d'une apostrophe,
- mais vous pouvez travaillez directement dans le dossier `tpchomp` (sans sous-dossier par exercice).

Exercice 1 – Introduction et téléchargement

Dans ce TP, vous allez utiliser la bibliothèque qtido (bien penser à récupérer la **dernière version** sur le site du cours).

NB : si vous écrivez un programme qui prend trop de temps, vous pouvez l'interrompre avec `Ctrl+C`.

Exercice 2 – Liste de listes

Pour cet exercice, travaillez dans un fichier `ex2.py`, directement dans le dossier `tpchomp`. Rappel : quand on parle ici de paramètres, ce sont des paramètres de fonctions.

Q1) Écrivez une fonction `creer_liste_zeros` qui crée une liste de longueur le nombre en paramètre contenant que des `0` et qui renvoit la liste créée. Dans le programme principal, appelez la fonction `creer_liste_zeros` pour créer une liste de longueur `10` et affichez la liste créée, élément par élément.

Q2) Écrivez une fonction `creer_liste_de_listes` qui prend un nombre en paramètre, crée une liste contenant comme seul élément une liste de `0` de taille le nombre passé en paramètre et renvoit la liste créée. Pour cela appelez dans `creer_liste_de_listes` la fonction `creer_liste_zeros` définie dans la question précédente. Dans le programme principal, appelez la fonction `creer_liste_de_listes` et récupérez l'élément de la liste créée. Affichez ensuite élément par élément le contenu de la liste.

Q3) Modifiez la fonction `creer_liste_de_listes` pour qu'elle renvoie une liste contenant deux listes de `0` de taille le nombre passé en paramètre. Dans le programme principal, affichez élément par élément le contenu de la liste.

Q4) Modifiez la fonction `creer_liste_de_listes` pour qu'elle prenne deux nombres `lignes`,`colonnes` en paramètre et qu'elle crée une liste contenante `lignes` listes de `0` de taille `colonnes` (cela correspond à créer un tableau de `lignes` lignes et `colonnes` colonnes). Dans le programme principal, appelez la fonction `creer_liste_de_listes` avec `3` et `7` comme paramètres et affichez en suite élément par élément la liste de listes renvoyée en spécifiant sa position dans le tableau.

Exercice 3 – Jeu du Chomp

Attention : pour cet exercice vous pouvez utiliser les fonctions de l'exercice précédent. Une documentation du module `qtido` est disponible à l'adresse : http://learn.heeere.com/2017-info-6a42/poly/reference-qtido/

Toutes les questions de cet exercice doivent être réalisées dans le fichier `chomp.py`, à créer.

Le but de cet exercise est d'implémenter le jeu du Chomp, dont les règles sont les suivantes : deux joueurs se disputent une tablette de chocolat de taille `n`,`m` (des entiers supérieurs à 1); chaque joueur choisit alternativement un carré de chocolat, le mange et mange aussi tous les carrés qui se trouvent en bas et à sa droite; la partie s'arrête quand un des deux joueurs mange le carré en haut à gauche.

Dans cet exercise, vous allez donc afficher à l'aide du module `qtido` une tablette de chocolat de 8x6 carrés et jouer au Chomp à la souris.

Q5) Faites en sorte que votre programme affiche une fenêtre de taille `800`,`600`, puis, à l'aide de la boucle `while` et de la fonction `est_fermee` du module `qtido`, affiche au terminal "en éxecution" en boucle en attendant la fermeture de la fenêtre par l'utilisateur.

Q6) Écrivez la fonction `dessiner_carre` qui prend en paramètre une fenêtre `f` et trois entiers `x`,`y`,`taille` et affiche un carré de couleur marron dont le point en haut à gauche est à la position (x,y) dans la fenêtre et les arêtes sont de taille `taille`. Appelez la fonction `dessiner_carre` dans votre programme pour qu'elle dessine un carré de taille 100 en haute à gauche dans la fenêtre.

Q7) Modifiez la fonction `dessiner_carre` pour que le carré dessiné ait un bord blanc (un cadre).

Q8) Modifiez votre programme pour qu'il affiche une ligne de carrés de chocolat (8 carrés) tout en haut de la fenêtre, à l'aide de la fonction `dessiner_carre`.

Q9) Modifiez votre programme pour qu'il affiche une tablette de chocolat (de 8x6 carrés).

Q10) Dans votre programme principal, générez un tableau de taille (8,6) à l'aide de la fonction `creer_liste_de_listes`. Ce tableau va representer les carrés de la tablette de chocolat et leur état (mangé ou pas) : l'élément à la place (i,j) du tableau va valoir 1 si le carré correspondant a été mangé (donc, il ne doit pas être affiché) et 0 s'il n'a pas été mangé (il va falloir donc l'afficher).

Q11) Ecrivez la fonction `carre_au_hasard` qui prend en paramètre un tableau et fixe un élément au hasard de ce tableau à `1`. Appelez cette fonction dans votre programme pour qu'à chaque itération de la boucle `while` un carré au hasard de la tablette soit mangé et la fenêtre affiche la tablette de chocolat sans les carrés qui ont été mangés jusqu'à ce moment. Attention : il faudra utiliser la fonction `effacer` de `qtido` avant de pouvoir redessiner la fenêtre.

Q12) A l'aide des fonctions `est_souris(f, e, "PRESS")` et `coordonnees_souris(f, e)` (nouvelles dans qtido), affichez les coordonées du point de la fenêtre sélectionné avec la souris. Pour cela, vous aurez besoin d'appeler aussi les fonctions `attendre_evenement` et `dernier_evenement` du module qtido. Attention : la fonction `coordonnees_souris` ne doit être appelée que si `est_souris` renvoie `True` (si on a fait un click avec la souris).

Q13) Ecrivez la fonction `carre_selectionne` qui, à partir des coordonnées d'un point `x`,`y` de la fenêtre, renvoit les indices du carré dans la tablette de chocolat sélectionné, sous forme d'une liste de deux entiers.

Q14) Ecrivez la fonction `manger_carres` qui prend en paramètre un tableau et deux entiers et qui fixe à `1` les éléments du tableau qui ont des indices superieurs ou égaux aux entiers passés en paramètre.

Q15) Modifiez votre programme pour qu'il arrête d'éliminer au hasard des carrés et pour qu'il mette à jour la tablette de chocolat à chaque click de souris.

Q16) (Optionnel) Modifiez le programme pour que la taille de la tablette de chocolat soit définie par l'utilisateur en passant le nombre de carrés par lignes et pas colonnes comme paramètres du programme. Ensuite, faites en sorte qu'à la fin de la partie le programme dise si c'est le joueur 1 ou le joueur 2 qui a gagné. Pour cela, il faudra vérifier si les cliques faits sont valides (sur un carré qui n'a pas encore été mangé).