

Atelier JPO - déroulement

Bienvenue dans cet atelier qui vous permettra d'en apprendre plus **sur Python** et particulièrement **Tkinter**, un module de base de Python.

Votre but est de compléter le code afin de reproduire le célèbre **jeu du dinosaure** de Google Chrome !

Prêt ? Alors c'est parti pour coder !

Tout d'abord, pour utiliser tkinter, il faut l'**importer** !

En langage Python, on importe un module et la totalité de ses composants en utilisant

```
from nom_du_module import *
```

- **Essayez d'importer le module tkinter !**

On importe également le module **random**, qui nous servira plus tard à générer des **éléments aléatoires**.

Par la suite, on met les dimensions de la fenêtre de jeu à 960 x 720 pixels, ce qui sera largement suffisant.

- Il faut alors initialiser la fenêtre de jeu à l'aide de Tkinter et de la fonction **Tk()**. Essayez alors d'affecter la fonction Tk() à la variable **game** à l'aide de cette fonction !

Une fois la fenêtre de jeu initialisée, il faut ajouter à celle-ci **le composant principal** qui accueillera tous les autres composants (dino, cactus, etc...). Cet élément se nomme **Canvas** en **tkinter** et prend une multitude de paramètres tels que :

- la fenêtre de jeu
- la largeur **width**
- la longueur **height**
- le fond de couleur (background) **bg**

- **A l'aide de ces informations, essayez d'instancier l'élément principal du jeu ! On le nommera frame.**

Indice: on utilisera la syntaxe Canvas (<fenêtre de jeu>, width = <dimension>, height = <dimensions>, bg = <'couleur'> ...)

Une fois l'élément principal initialisé, il faut initialiser **tous les autres composants** dont on a besoin ! Dans le cas de notre jeu, nous avons :

- 2 images de dinosaures
- 1 image de sol
- 4 différentes images de cactus
- une chaîne de caractères pour le score

Pour instancier une image, on utilise la méthode `PhotoImage(<chemin du fichier>)` ainsi que les méthodes `zoom()` et `subsample()` pour ajuster la taille de celle-ci. Ces méthodes créent une nouvelle image modifiée avec les traitements d'image appliqués.

- Ainsi, essayez d'initialiser l'image '**cactus.png**' en faisant un **zoom de 7** et un **traitement d'image subsample de 5** !

Indice: Aidez vous des cactus suivants pour la syntaxe

Passons désormais à la création du score. Pour cela, on va créer une variable **Score** qui ajoutera un texte sur notre élément principal **frame** créé précédemment.

- A l'aide de la méthode **create_text(width,height)** de frame, créer le texte lié au score, pas encore fonctionnel pour le moment.

Maintenant, nous voulons afficher le score dynamiquement afin qu'il augmente selon la progression du joueur.

Pour ce faire, nous allons créer une méthode récursive "**incrimente_score_joueur**" qui sera appelée toutes les 100 millisecondes.

- Dans la fonction "incrimente_score_joueur", incrémentez y la variable "**score_joueur**" puis utilisez la méthode **after()** de la variable **frame** afin de mettre en place la récursivité, avec en paramètre : le nombre de millisecondes et la méthode à appeler (en l'occurrence, elle même).

Beaucoup de fonctions servent à rendre le jeu esthétiquement plus sympathique mais n'affectent en rien la jouabilité, comme par exemple **next_image()** qui va gérer l'image du sol se répétant à l'infini. Ce n'est pas le cas des fonctions **apparition_cactus()** et **move_cactus()** qui génèrent et déplacent les cactus vers le dinosaure. Vous pouvez jouer avec les valeurs de ces fonctions afin de modifier la difficulté du jeu comme bon vous semble!

Les fonctions "**dinoMonte()**" et "**dinoDescend()**" gèrent la position du dinosaure selon sa hauteur. Il nous manque donc seulement l'implémentation du saut, que nous lierons à une touche.

- Pour cela, utilisez la méthode **bind** de la variable **game**, qui va lier une action à une fonction.

Indice: La syntaxe se décompose en 'bind("<KeyPress-space>", fonction)' pour lier le saut sur espace

Et voilà! Vous avez désormais un jeu du dinosaure fonctionnel et modulable selon vos envies!