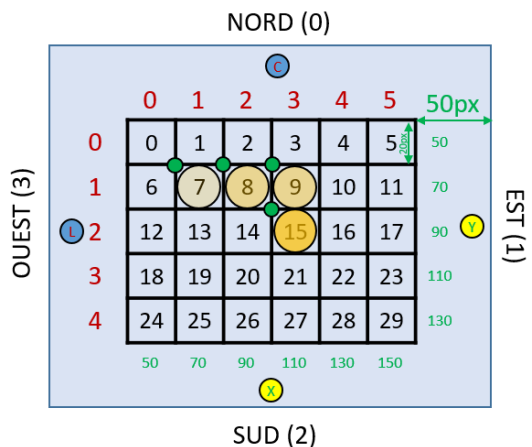


Aide au jeu du SNAKE en Python procédural avec le module `tkinter.py`

Principes généraux

Déplacement d'un serpent sur une grille virtuelle dont chaque case est repérable de 3 manières différentes : **le numéro de la case dans l'ordre gauche/droite et haut/bas**, les **numéros de ligne et de colonne de la case**, les **coordonnées x et y du coin supérieur gauche de la case**.



Position de la tête du serpent

- Case n°15
- Coordonnées Ligne, colonne (2, 3)
- Coordonnées pixel haut gauche (110, 90)

Position de la queue du serpent

- Case n°7
- Coordonnées Ligne, colonne (1, 1)
- Coordonnées pixel haut gauche (70, 70)

Direction serpent : SUD

Marges de la zone de déplacement du serpent : 50px des 4 côtés

Taille des cases de la zone de déplacement : 20px sur 20px

Représentation du serpent : [(15, ①), (9, ②), (8, ③), (7, ④)]

Identifiants des objets "create_oval" du Canvas

Contrainte imposée

Utilisation du fichier `snake_pattern.py`, respect des fonctions et de la structure du programme proposées, organisation des éléments du programme et ajouts dans les sections dédiées.

Section Importations

Il s'agit de toutes les importations nécessaires, vous devriez avoir besoin au minimum des fonctions du module `tkinter`, de la fonction `sleep` du module `time` et d'une des fonctions `randint`, `randrange` ou `choice` du module `random`.

Section constantes du monde

Il s'agit des variables qui ne sont plus modifiées une fois le programme lancé. Elles sont toujours écrites `EN_MAJUSCULE`. Le programme ne doit contenir aucune de ces valeurs « en dur » (couleurs, tailles, directions...). Vous pourriez proposer par exemple `T_CASE` (taille en pixel d'une case de la grille de déplacement du serpent), `NB_COL` (nombre de cases de la

grille), `NB_LIG` (nombre de lignes de la grille), `NORD`, `EST`, `SUD`, `OUEST` (les valeurs respectivement de 0, 1, 2, 3 pour représenter les 4 directions possibles du serpent)...

Section variables globales du serpent

Il s'agit des variables qui seront modifiées pendant le cours du jeu. Au minimum il s'agit de la `direction` courante du serpent et de la variable `snake` qui permet de représenter le serpent à tout instant.

Section variables globales de l'interface

Il s'agit au minimum de la fenêtre principale de l'application et de plusieurs dictionnaires qui permettront de regrouper tous les éléments de l'interface par type en leur affectant une clé sous la forme d'une chaîne de caractère qui évoque son utilisation. Ces dictionnaires seront vides au départ puis « peuplés » par des fonctions adéquates associées à chaque `Frame` de l'interface.

Exemple d'appel au bouton quitter pour le griser :

```
tk_boutons['quitter'].configure(state=DISABLED)
```

Section fonctions de calcul sur les positions et la direction du serpent

- 6 fonctions permettant de transposer une des 3 manières (ligne/colonne, pixels, numéro de case) de repérer une case vers une autre
- La fonction de calcul de la $n^{\text{ème}}$ case suivante d'une case donnée en fonction de la direction (sur un tore → la case qui suit la dernière ligne (resp. colonne) est la première ligne (resp. colonne))
- La fonction de calcul de la nouvelle direction après avoir pivoter de $n * 90^\circ$ dans le sens horaire

Section des callbacks

Il s'agit des fonctions qui réagissent à un événement utilisateur. Les événements liés à un click bouton sont quitter et reset. Les autres sont liés aux événements click gauche et droit de la souris ou flèches gauche et droite du clavier.

Section de la construction graphique du jeu

Chaque fenêtre est sous-fenêtre a une fonction associée qui permet de définir les éléments de l'interface qu'elle contient (ces éléments sont sauvegardés en les regroupant par type dans des dictionnaires globaux)

La construction de l'interface se fait en deux parties. La première construit les éléments qui ne seront plus modifiés puis appelle une fonction qui initialise l'interface (cette seconde partie autonome pourra ainsi être rappelé pour commencer une nouvelle partie)

Section initialisation des variables globales

Il s'agit d'initialiser les variables globales du serpent

Section de la construction du serpent

Une fonction construit le serpent initial et une autre provoque son déplacement de 1 case dans la direction courante (astuce : un déplacement consiste à déplacer la queue dans la position qui suit la tête dans la direction donnée)

Fonctionnalités supplémentaires (liste ni obligatoire ni exhaustive)

1. Evolution du score (+1 à chaque déplacement du serpent)
2. Gestion d'apparition d'obstacles
3. Gestion de mort par contact de la tête avec un obstacle ou une partie du serpent
4. Gestion de l'agrandissement du serpent en fonction du score (+n chaque tranche de score)
5. Gestion de l'apparition de nourriture et de l'agrandissement par contact de la tête et de la nourriture (+ évolution du score)
6. Gestion de niveaux (taille serpent, vitesse, obstacles, objectifs...)
7. Gestion d'un jeu à 2 serpents (1 dirigé à la souris et l'autre au clavier)