



BRANCHE	SECTION(S)	ÉPREUVE ÉCRITE
INFORMATIQUE	1B	Durée de l'épreuve : 3h Date de l'épreuve :

Dans la suite, vous allez écrire l'application dénommée « **splitit** ». Il s'agit d'un petit jeu qui a pour but d'éliminer des disques respectivement des carrés en cliquant dessus. À chaque fois qu'on clique sur une figure, celle-ci se divise en deux. Le jeu continue jusqu'à ce les figures soient devenues trop petites et disparaissent.

Créez l'application **splitit.py** dans le dossier qui vous a été indiqué (p. ex. : LXY\_0123). Ce dossier sera appelé dossier de travail dans la suite. Sauvegardez votre fichier dans ce dossier !

### Initialisations diverses

[2 pts]

- Les seules importations permises sont celles de **pygame**, **pygame.locals**, **sys**, et des fonctions **randint** du module **random** et **sqrt** du module **math**. [1 pt]
- Définissez dans votre programme les variables globales suivantes : [1 pt]
  - width** correspond à la largeur de la fenêtre de jeu et vaut 500 pixels ;
  - height** correspond à la hauteur de la fenêtre de jeu et vaut 400 pixels ;
  - min\_size** et **max\_size** représentent les dimensions (rayon pour le disque, demi-côté pour le carré) minimales et maximales que la figure peut avoir. Ils sont initialisés respectivement aux valeurs 10 et 40 pixels ;
  - start\_size** représente la dimension initiale de la figure. Elle est initialisée à 40 pixels ;
  - split\_size** représente la dimension en-dessous de laquelle une figure n'est plus divisée et disparaît. Elle est initialisée à 6 pixels ;
  - time** représente le temps de jeu disponible, initialisé à 60 secondes ;
  - points** représente le nombre de points obtenus, initialisé à 0 points.

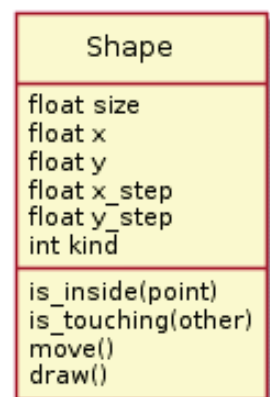
### La classe Shape

[15 pts]

- La classe **Shape** représente une figure géométrique (carré ou disque) de dimension **size**. Le centre de la figure a pour coordonnées **x** et **y**. Les attributs **x\_step** et **y\_step** déterminent le sens de mouvement de la figure (valeur positive → vers la droite/vers le bas, valeur négative → vers la gauche/vers le haut). L'attribut **kind** représente le type de figure (0 = disque, 1 = carré). [1 pt]

Le constructeur initialise la dimension, les coordonnées, les sens de mouvement et le type sur les valeurs passées aux paramètres.

- La méthode **is\_inside** vérifie si le point passé en paramètre se trouve à l'intérieur de la figure. Un point est par définition un tuple (**x**, **y**) où **x** et **y** sont des réels. [3,5 pts]



- e) La méthode **is\_touching** vérifie si la figure actuelle et la figure passée en paramètre sont entrées en collision. Si les deux figures sont des carrés, il faudra vérifier si la distance verticale et horizontale des deux centres sont toutes les deux inférieures à la somme des deux demi-côtés. Si l'une au moins des deux figures est un disque, on vérifiera si la distance de leurs centres est inférieure à la somme de leurs dimensions. [3 pts]
- f) La méthode **move** ajoute un nombre aléatoire entier entre -2 et +2 (bornes incluses) à la dimension de la figure à condition que la dimension résultante se situe entre les limites **min\_size** et **max\_size**.  
Elle vérifie ensuite si la figure a atteint l'un des bords de la fenêtre d'application : dans ce cas, la figure doit rebondir correctement vers l'intérieur de la fenêtre (en modifiant le signe de **x\_step** ou **y\_step**). Elle ajoute ensuite les valeurs de **x\_step** (respectivement de **y\_step**) aux attributs **x** respectivement **y**. [5 pts]
- g) La méthode **draw** dessine la figure de centre (**x**, **y**) sous forme d'un disque de rayon **size**, respectivement d'un carré de demi-côté **size**. Chaque figure a un bord noir d'épaisseur 1 pixel. [2,5 pts]

### **La classe Shapes**

**[19 pts]**

- h) La classe **Shapes** gère une liste de figures de type **Shape**. A cet effet elle dispose d'un seul attribut qui est la liste **shapes\_list**. Le constructeur initialise la liste à une liste vide. [0,5 pts]
- i) La méthode **size** retourne le nombre d'éléments de la liste. [0,5 pts]
- j) La méthode **add\_random** aux paramètres **x**, **y** et **size** ajoute aléatoirement un disque ou un carré de centre (**x**, **y**) et de dimension **size** à la liste **shapes\_list**. Les valeurs des attributs **x\_step** respectivement **y\_step** sont définies par un nombre aléatoire à 2 décimales situé entre -5.00 et +5.00 (bornes comprises). [3,5 pts]
- k) La méthode **reset** vide la liste et ajoute une figure aléatoire à la liste (utilisez la méthode définie précédemment). La dimension correspond à **start\_size**. Les coordonnées du centre sont aléatoires, mais la figure doit se situer complètement à l'intérieur de la fenêtre de jeu. [2,5 pts]
- l) La méthode **split** prend comme paramètre un indice correspondant à une figure de la liste **shapes\_list**. Elle vérifie tout d'abord si en divisant la dimension de la figure par 2, le résultat est supérieur ou égal à **split\_size**. Dans ce cas, la méthode ajoute à la liste deux nouvelles figures aléatoires ayant pour dimension la moitié de la dimension actuelle.  
Les centres des nouvelles figures doivent avoir la même ordonnée que la figure initiale et leurs abscisses seront choisies de sorte que les nouvelles figures soient « adjacentes » (c'est-à-dire un bord commun dans le cas de deux carrés, un seul point commun dans le cas de deux disques par exemple).  
La méthode efface ensuite la figure actuelle de la liste, peu importe si de nouvelles figures ont été ajoutées ou non. Ainsi les figures devenues trop petites disparaissent. [3 pts]
- m) La méthode **get\_clicked\_shape** recherche dans la liste **shapes\_list** l'indice de la figure pour laquelle le point **p** (tuple) passé en paramètre se trouve à l'intérieur de la

figure. Dans le cas où aucune figure vérifiant la condition n'est trouvée, la méthode retourne -1. [6 pts]

- n) La méthode **move** appelle pour chacune des figures de la liste **shapes\_list** la méthode correspondante de la classe **Shape**. [1,5 pts]
- o) La méthode **draw** dessine chacune des figures de la liste **shapes\_list**. [1,5 pts]

**Programme principal - Initialisations** [2 pts]

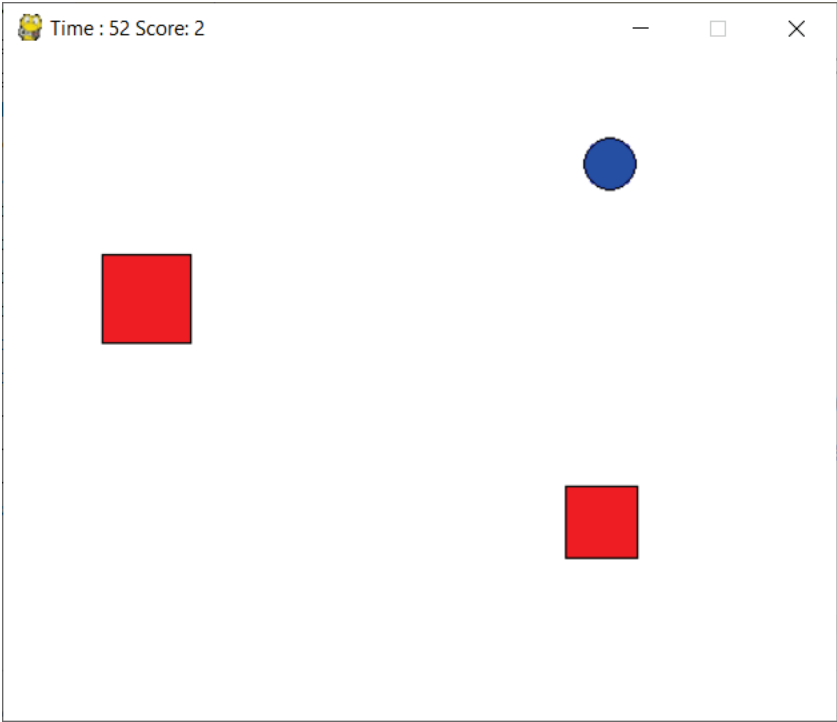
- p) La fenêtre d'application a pour dimensions **width** et **height**. La surface de dessin est rafraîchie 25 fois par seconde. Créez une instance **my\_shapes** de la classe **Shapes**. Initialisez **my\_shapes** à l'aide de la méthode **reset**.

**Boucle principale - Réaction aux évènements** [7 pts]

- q) Lorsque l'utilisateur ferme la fenêtre, le programme se termine proprement. [1 pt]
- r) Lorsqu'on frappe la touche "n", le jeu recommence. [3 pts]
- s) Un clic sur un bouton de la souris divise la figure à l'intérieur de laquelle on vient de cliquer et le nombre de points est incrémenté de 1. Si aucune figure n'a été cliquée, rien ne se passe. [3 pts]

**Boucle principale - Actions répétées à chaque itération** [15 pts]

- t) On vérifie pour chaque figure de la liste si elle entre en collision avec une figure du reste de la liste. Dans ce cas, le mouvement horizontal des deux figures est inversé. Les figures sont ensuite mises en mouvement jusqu'à ce qu'elles ne se touchent plus. [6 pts]
- u) Tant que le jeu n'est pas terminé : [4 pts]
- la fenêtre est repeinte en blanc ;
  - l'entête de la fenêtre d'application est "Time:" suivi du temps restant sous forme d'un nombre entier, du texte "Score:" ainsi que du nombre de points obtenus ;
  - toutes les figures sont dessinées et mises en mouvement.
- v) Si le temps n'est pas écoulé, on retranche 1/25 du temps restant, sinon le jeu est terminé et l'entête de la fenêtre d'application affiche "You lost!". Toutes les figures sont arrêtées et un clic sur une des figures ne produira plus aucun effet. [3 pts]
- w) S'il n'y a plus de figures, le jeu est terminé et l'entête de la fenêtre d'application affiche "You won!". [2 pts]



Jeu en cours d'exécution