LE GOUVERNEMENT
DU GRAND-DUCHÉ DE LUXEMBOURG
Ministère de l'Éducation nationale,
de l'Enfance et de la Jeunesse

EXAMEN DE FIN D'ÉTUDES SECONDAIRES CLASSIQUES
2019

BRANCHE	SECTION(S)	ÉPREUVE ÉCRITE
Informatique	**B**	*Durée de l'épreuve :* *180 minutes* *Date de l'épreuve :*

Créez le programme **Fireworks.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 !

Le programme **Fireworks.py** est basé sur les classes **`Particle`** et **`Firework`** et fait appel à la librairie **`pygame`**. Il simule un feu d'artifice avec des charges rouges, vertes, bleues et blanches. Une charge qui explose crée un nombre variable de particules lumineuses qui tombent vers le bas et qui s'éteignent après un certain temps.

Afin de vous familiariser avec le fonctionnement du programme, vous trouverez une version exécutable (**`demo_fireworks`**) dans le sous-dossier **`demo`** de votre dossier de travail.

I) Initialisation (1 p.)

Effectuez le chargement des bibliothèques nécessaires. (1 p.)

II) `Firework` La classe Particle : (18 p.)

1) La classe **`Particle`** décrit une particule lumineuse. Elle est décrite par les attributs suivants :
   - x, y : les coordonnées de son centre (float),
   - dx, dy : les composants du vecteur vitesse (float),
   - color : la couleur (rouge, verte, bleue ou le blanche)
   - size : la taille,
   - trail : la liste de couples de coordonnées (tuples) pour la trainée (*en allemand : Schweif*)

   Les particules blanches diffèrent des particules colorées par leur affichage (voir la méthode **`draw`** ci-dessous).

2) Le **`constructeur`** initialise les coordonnées, les composants du vecteur vitesse et la couleur selon les valeurs passées par paramètres. La valeur par défaut des composants du vecteur vitesse est 0. La couleur par défaut est le blanc. La trainée est initialisée sur le point de création. Ensuite les composants du vecteur vitesse sont ajustés : **`dx`** est incrémenté d'une valeur entière aléatoire dans [-8, 8], **`dy`** d'une valeur entière aléatoire dans [-8, 0]. La taille est une valeur entière aléatoire dans [10, 15]. (5 p.)

3) La méthode **`draw`** dessine la particule sur la toile de **`pygame.`**
   a) Une particule colorée est formée par disque de rayon **`size`**, centré en **`(x,y)`**, et une trainée en forme de ligne polygonale d'épaisseur 1px reliant les points contenus dans la liste **`trail`**. N.B. : Les paramètres x, y et rayon de la méthode circle n'acceptent que des arguments de type int.
   b) Une particule blanche n'a pas de traînée. Elle est formée d'une croix (**X**) de taille **`size`** x **`size`**, centrée en **`(x,y)`**, et d'épaisseur 1px. Pour créer un effet de scintillement pour les particules blanches, celles-ci ne sont dessinées qu'aléatoirement (avec une probabilité de 50%) et leur affichage est en plus décalé aléatoirement dans les deux directions (horizontale et verticale) d'une amplitude entière aléatoire dans [-2, +2]. Les attributs x et y de la particule ne devront pas changer. N.B. : Les paramètres start_pos et end_pos de la méthode line acceptent des arguments de type float. (6 p.)

4) La méthode **`move`** déplace la particule de **`dx, dy`** pixels. En cas de collision du centre **`(x,y)`** avec un côté de la fenêtre d'application, le mouvement est à inverser, de façon à ce que la particule ne sorte jamais complètement de la fenêtre. Après le mouvement, la taille est réduite de 0.2, le vecteur vertical est incrémenté de 0.4 (c'est l'effet de la gravitation). Ensuite la nouvelle position est ajoutée à la trainée. La taille de la trainée est à limiter aux 20 points les plus récents. (7 p.)

III) La classe Firework : (19 p.)

La classe **`Firework`** gère une liste de particules de type **`Particle`**. A cet effet elle dispose d'un seul attribut qui est la liste **`particle_list`**.

1) Le **`constructeur`** initialise la liste vide **`particle_list`**. (2 p.)
2) La méthode **`add_cracker`** aux paramètres **`x`**, **`y`**, **`dx`**, **`dy`** et **`kind`** crée et ajoute 60 particules colorées ou 120 particules blanches à la liste, toutes ayant comme centre **`(x,y)`** et comme composants du vecteur vitesse **`dx`** et **`dy`** : ces particules constituent par définition une charge du feu d'artifice. Le paramètre **`kind`** indique le type de la charge : si **`kind`** = 1 alors toutes les particules sont colorées (d'une même couleur aléatoire parmi le rouge, le vert ou le bleu), si kind = 2 alors toutes les particules sont blanches. (4 p.)
3) La méthode **`draw`** dessine sur la toile **`pygame`** toutes les particules de la liste. (2 p.)
4) La méthode **`move`** effectue un déplacement de toutes les particules de la liste. (2 p.)
5) La méthode **`boost`** fait remonter toutes les particules ***non blanches*** de la liste en réduisant de 10 (dix) leur composant vertical du vecteur vitesse. (2 p.)
6) La méthode **`purge`** supprime toutes les particules de la liste dont la taille est strictement inférieure à 2. Avant d'être supprimée, une particule ***colorée*** peut (avec une probabilité de 1%) donner naissance à une nouvelle charge de type 1 qui est à ajouter à la liste. Les centres des nouvelles particules coïncident avec celui de la particule à supprimer, leurs composants du vecteur vitesse sont choisis égaux à 0. (5 p.)
7) La méthode **`clear`** supprime toutes les particules de la liste. (2 p.)

IV) Le programme principal (22 p.)

1) Effectuez les initialisations nécessaires pour créer une fenêtre d'application avec une surface de dessin de 1000 x 700 pixels et dont l'entête sera 'Fireworks – LYCEE – NUMERO'. Finalement une instance de Firework est créée. Initialisez aussi des variables pour mémoriser les coordonnées de la souris et les composants du vecteur vitesse. (4 p.)
2) La boucle principale du programme effectue 25 fois par seconde la gestion complète des particules (déplacement, suppression, réaffichage complet) et gère les actions de la souris et du clavier : (3 p.)
   a) Lorsqu'on **enfonce** un bouton de la souris, on mémorise le bouton et les coordonnées. (3 p.)
   b) Lorsqu'on **déplace** la souris, on calcule les composants du vecteur vitesse (distance de la position actuelle de la souris par rapport au point de coordonnées mémorisées ci-dessus). Ces deux composants sont ensuite tous les deux divisés par 10. (2 p.)
   c) Lorsqu'on **relâche** le **bouton gauche** (resp. **droit**) de la souris, on ajoute aux coordonnées de la souris et avec les vecteurs réduits (voir ci-dessus) une charge **colorée (**resp. **blanche)** à la liste. (4 p.)
   d) Lorsqu'on appuie sur la touche **« s »** du clavier, toutes les particules de la liste sont supprimées. (1 p.)
   e) Lorsqu'on appuie sur la touche **« b »** du clavier, on booste toutes les particules colorées de la liste. (1 p.)
   f) Lorsqu'on appuie sur la touche **« a »** du clavier, on active/désactive le mode automatique du programme. En mode automatique, on ajoute chaque seconde une charge de type aléatoire avec des coordonnées aléatoires et des vecteurs nuls dans le quart supérieur de la fenêtre. Ce mode automatique est désactivé au départ du programme. (1+3=4 p.)
   g) Un clic sur la case de fermeture, ou l'appui de la touche ESCAPE du clavier permettent de quitter l'environnement **`pygame`** et de terminer l'application à tout moment. (1 p.)