Programmer un jeu vidéo avec Pyxel : 3/6

### Ajouter des ennemis

Nous repartirons dans ce tutoriel du script réalisé précédemment :

Faire Enregistrer Sous pour renommer à présent ce script en

# Pour démarrer…

**Rappels vus lors du tutoriel précédent :**

* Comment définir une liste vide ? ……………………………
* Comment parcourir une liste en balayant chaque élément ? ……………………………………….
* Par quoi était représenté un tir dans la mémoire de l’ordinateur ? ……………………………………………….. ……………………………………………………………………………………………………………..
* Comment ajouter un élément à la fin de  ? ……………………………………………..
* Qu’est-ce qui déclenchait l’ajout d’un tir dans la liste ? ……………………………………………………………..
* Comment étaient déterminées les coordonnées du nouveau tir ? ………………………………………………..
* Quelle instruction faisait se déplacer le tir ? ……………………………………………………..
* Quel module Python contient les fonctions utilisant de l’aléatoire ? ………………………………………….

Cette étape ressemblera beaucoup à la précédente, avec deux différences cependant : les ennemis apparaîtront à un **endroit aléatoire**, et à **intervalles de temps régulier** (sans action de l’utilisateur)

# Ajouter des ennemis

## Créer la liste des ennemis, et dessiner les ennemis

Chaque ennemi sera représenté par un carré rose de côté 8 pixels. Au début du jeu, la liste des ennemis sera vide. A intervalles de temps régulier, un nouvel ennemi sera généré et ajouté à la liste. Comme pour les tirs, le carré sera stocké en mémoire par les coordonnées de son coin supérieur gauche.

### **Dessin**

Dans cette partie, on suppose que la liste des ennemis est déjà créée.

Pour les dessiner à l’écran, on va balayer en appelant l’élément rencontré au fur et à mesure, et on utilisera en lui passant les coordonnées de cet ennemi

Dans quelle fonction se trouvera cette boucle dessinant les tirs ? ……………………………….

Ecrire ci-contre le corps de cette boucle :

**Création de la liste d’ennemis**

Dans le cas des ennemis, on veut les faire apparaître **en haut** de l’écran : comment traduire cela sur les coordonnées du tir ? ……………………………

On veut aussi que l’ennemi puisse apparaitre n’importe où en haut de l’écran. On va donc choisir une valeur aléatoire (entière) pour *x* qui soit comprise entre ………… et ………………………………………….. Pour cela, on utilisera la fonction Python qui renvoie un nombre entier aléatoire entre a et b inclus. Cette fonction se trouve dans le module random, qu’il faut importer tout comme pyxel.

Ecrire l’instruction qui ajoute un ennemi à la liste : ……………………………………………………………………………….

On veut qu’un nouvel ennemi apparaisse chaque seconde. Rappelons que la fonction est appelée 30 fois par seconde, et chacun de ces appels constitue ce qu’on appelle un « frame ». Pyxel donne accès au nombre de frames écoulé depuis le début du jeu grâce à la variable

Au bout de combien de frames doit-on créer le 1er ennemi ? …………… Le 2nd ? ………….. Le 3ème ……….

Il s’est écoulé 1680 frames : doit-on créer un ennemi ? ……………………….. Même question pour 3745 frames ………………………………………………………………….

Comment tester dans le cas général s’il est temps de créer un nouvel ennemi ? …………………………………… …………………………………………………………………………………………………………………………………………………………….

Compléter alors la fonction ci-dessous qui sera appelée par



|  |
| --- |
| **Jalon 1** : quand on lance le script, on doit voir le vaisseau au centre de l’écran avec des ennemis qui apparaissent chaque seconde en haut. Pour l’instant, ces ennemis sont fixes. |

## Déplacement des ennemis

On voudrait maintenant que les différents ennemis se déplacent vers le bas au fur et à mesure, pour créer l’animation. Pour cela, on va compléter la fonction en rajoutant un appel à une fonction

Voici une ébauche cette fonction



* Quel est le type du ***paramètre formel*** de cette fonction *(celui qui figure dans la parenthèse)* ? ………………………………………………………………………………….
* Lors de son utilisation, quel ***paramètre effectif*** lui sera passé ? *(celui avec lequel on fait réellement « fonctionner » la fonction lors de l’exécution du script*) ………………………………………………
* L’élément courant de la liste est la variable (au singulier) : quel est son type ? ……………..……………………………………………………………………………………………………………………..
* Comment tester si le carré matérialisant le tir est entièrement sorti du cadre par le haut ? …………………………………………………………………………………………..
* Compléter les lignes du script, et intégrer ce code dans votre script de jeu.

|  |
| --- |
| **Jalon final** : Les ennemis apparaissent et se déplacent. Pour le moment, les tirs n’ont pas d’effet… |