CHANEMOUGAM Kaveen

TEXIER Nathanael

Compte-rendu du projet en Python.

Nous avions pour ce semestre un projet en Python à réaliser et notre choix s'était directement porté sur la création d'un jeu très connu, le fameux « Space Invaders ». Pour cela, nous avions opté pour l'utilisation du module PyGame de Python car les modules TKinter et Turtle vus en classe était bien trop « faible » pour ce projet, mais aucun de nous deux ne savait l'utiliser. Durant l'intégralité des vacances de Toussaint, nous nous sommes donc formés sur ce module à travers différents exercices et explications trouvés sur internet. Malgré tout, nous pensions ne pouvoir jamais terminer le programme à temps. Nous avons donc cherché différents projets déjà faits sur internet mais il était impossible pour nous de s'en inspirer étant donné qu'ils utilisaient tous les « classes » de Python, chose que nous ne connaissions guère.

Durant les vacances, nous avons donc cherché les différentes images nécessaires pour la création du jeu, ce qui n'a pas été aussi simple que prévu à cause des fausses images « png » que l'on trouve sur internet. Nous avons donc découpé nous-même les images trouvées à l'aide d'un logiciel, rajoutant ainsi une difficulté supplémentaire à ce projet. Enfin, à la fin des vacances, nous avons appris à faire un fond pour notre jeu et à y mettre un vaisseau qui pouvait avancer horizontalement.

Nous avons commencé à réfléchir sérieusement seulement lors de la première séance accordée le Jeudi matin de la rentrée des vacances de Toussaint. En effet, la difficulté supplémentaire de ne pas pouvoir s'inspirer d'un projet déjà fait nous a vraiment rendu la tâche beaucoup plus complexe. Nous devions donc faire notre tout premier jeu, sans pouvoir s'inspirer d'un exemple, ce qui nous paraissait impossible. Mais abandonner ne pouvait pas être une solution pour nous. Au fil des séances que vous nous avez accordés, nous avons utilisé les divers conseils que vous nous avez donnés ainsi que des forums sur internet dans lesquels plusieurs personnes proposaient des morceaux de code, que nous utilisions ensuite si cela correspondait à nos besoins. En effet, nous n'utilisions pas tout le code donné mais seulement les lignes qui nous paraissaient intéressantes, que l'on modifiait par la suite pour les adapter à notre programme.

Ensuite, nous avions eu plusieurs problèmes au cours de notre projet. En effet des choses qui nous semblaient simple comme définir les limites de l'écran, créer un missile ou bien même tout simplement centrer le vaisseau en bas de l'écran ne furent pas aussi simple que prévu. Trouver comment faire sur internet fut assez compliqué mais nous avions réussi grâce aux quelques conseils que vous nous aviez donnés en classe. Puis un autre problème est apparu, celui du mouvement des ennemis et de leurs tirs aléatoires. Pour cela, chercher sur internet n'avait pas de sens pour nous car tous les programmes pour ce jeu étaient réalisés avec des classes, ne fonctionnant donc pas de la même manière pour notre programme. Nous avons donc utilisé un compteur (pour le déplacement horizontal) qui

augmentait en même temps que l'ennemi avançait, puis un deuxième compteur (pour le déplacement vertical) qui augmentait seulement lorsque l'ennemi descendait vers le bas. Ces deux compteurs ont donc été ajustés manuellement à travers divers tests car malgré après avoir effectué plusieurs recherches, nous n'avions trouvé aucune solution pour automatiser le déplacement des ennemis, nous obligeant ainsi une nouvelle fois à nous adapter en trouvant une solution nous-même.

Voilà ci-dessous le code que nous avons créé pour le déplacement d'un des ennemis :

```
#DEPLACEMENT ENNEMI 1
for position_ennemi in ennemis:
    if compteur_cote < 290  :
        position_ennemi.right += 1
        compteur_cote += 1
    elif compteur_cote == 290:
        position_ennemi.bottom += 40
        compteur_bas += 1
        compteur_cote += 1

    if compteur_bas == 1 and compteur_cote < 590:
        position_ennemi.right -= 2
        compteur_cote += 1
    elif compteur_cote == 590:
        position_ennemi.bottom += 40
        compteur_bas += 1
        compteur_cote += 1

    if compteur_bas == 2 and compteur_cote < 896:
        position_ennemi.right += 2.5
        compteur_cote += 1
    elif compteur_cote == 896:
        position_ennemi.bottom += 40
        compteur_bas += 1
        compteur_cote += 1

    if compteur_bas == 3 and compteur_cote < 1095:
        position_ennemi.right -= 3
        compteur_cote += 1
    elif compteur_cote == 1095:
        position_ennemi.bottom += 40
        compteur_bas += 1
        compteur_cote += 1

    if compteur_bas == 4 and compteur_cote < 1294:
        position_ennemi.right += 3.5
        compteur_cote += 1
    elif compteur_cote == 1294:
        compteur_bas += 1
```

Puis par la suite, un autre problème est apparu durant la création de notre programme, celui des erreurs dues à la collision entre nos missiles et les tirs ennemis ainsi que l'ennemi. En effet, il pouvait y avoir un cas rare où notre missile touchait deux tirs ennemis ou bien un tir ennemi et un ennemi en même temps, créant ainsi une erreur étant donné que notre missile devait disparaître à la suite d'une seule collision mais deux en avaient été provoquées en même temps.

A ce moment-là, notre programme était le suivant (avant que l'erreur soit corrigée) :

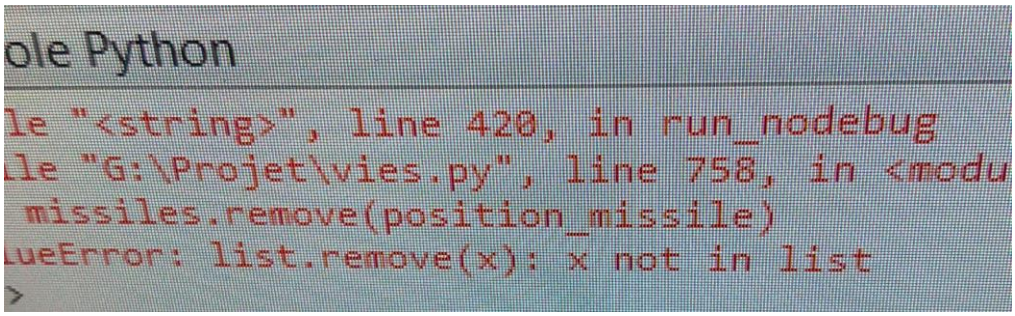
-Pour les collisions entre notre missile et le tir de l'ennemi :

```
if len(tirs) > 0 and len(missiles) > 0:
    collision_tir_missile = position_missile.colliderect(position_tir_ennemi)
    if collision_tir_missile == 1:
        tirs.remove(position_tir_ennemi)
        missiles.remove(position_missile)
```

Une condition « if » a donc été rajoutée à l'avant dernière ligne pour résoudre cette erreur:

```
if len(tirs) > 0 and len(missiles) > 0:
    collision_tir_missile = position_missile.colliderect(position_tir_ennemi)
    if collision_tir_missile == 1:
        tirs.remove(position_tir_ennemi)
        if missiles != []:
            missiles.remove(position_missile)
```

Cela nous permettant ainsi d'éviter ce type d'erreur :



Pour finir, après toutes les séances que vous nous avez accordées ainsi que plusieurs jours et heures de programmation chez-soi, nous avons enfin réussi à terminer notre projet, ce que nous pensions comme étant impossible. Le travail d'équipe est fortement utile dans ce genre de projet car en effet, nous avons remarqué que nous avancions beaucoup plus vite et efficacement lorsque l'on s'y mettait à deux. Les idées de l'un permettaient à l'autre d'avancer et réciproquement. Il nous faudrait beaucoup plus que 2 ou 3 pages pour expliquer toutes les difficultés rencontrées mais nous avons mentionné ici, celles qui nous ont posé le plus de problèmes.

Bien évidemment, nous avons réussi à finir ce projet du mieux qu'on le pouvait avec notre niveau. Nous sommes vraiment contents de nous mais nous savons très bien que ce jeu n'est pas parfait et qu'il est largement optimisable.

Enfin, voici l'évolution de notre projet initial jusqu'à notre projet final en quelques images :