

Pingame



Barré Aurélie

Geoffroy Chloé

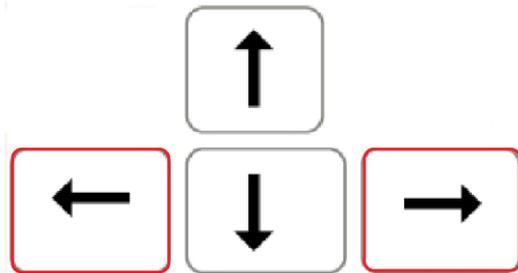
Sommaire

- I. Présentation
 - 1. Règles du jeu
 - 2. Choix du jeu
 - II. Outils utilisés
 - III. Répartition des tâches
 - IV. Problème rencontré
 - V. Programmation
 - VI. Le site Web
- Conclusion

I. Présentation

1. Règles du jeu :

Le but de ce jeu est d'exploser le maximum de météorite avec les punaises envoyées depuis le vaisseau sans se faire toucher. Attention vous n'avez que trois vies. Pour déplacer le vaisseau, utiliser les flèches gauches, droites de votre clavier. Des vies seront envoyées de temps en temps.



2. Choix du jeu :

Nous avons décidé de faire un type de space games mais de le modifier un peu. Par exemple, notre jeu lance les punaises automatiquement alors que le vrai space games envoie les projectiles avec la barre d'espace. Nous avons aussi choisi plusieurs vaisseaux et plusieurs météorites. Nous avons décidé de sélectionner un seul de chaque et qui soit visible sur notre fond d'écran.

Notre fond d'écran :



La météorite choisie :



Le vaisseau choisie :



Le projectile :



II. Outils utilisés

Nous avons utilisé tkinter. Tkinter permet de créer des interfaces graphiques et tkinter est une bibliothèque de Python.

Nous avons aussi utilisé la bibliothèque random. Cette bibliothèque permet la génération de nombres aléatoires. (ex : les missiles descendent aléatoirement)

Move permet de déplacer une image.

Bind_all permet d'associer une touche du clavier avec une fonction.

After permet de gérer le temps en millisecondes.

III. Répartition des tâches

Au tout début nous avons programmé ensemble pour se mettre d'accord sur les images, leurs positions et la vitesse du vaisseau, du missile et des punaises. À partir de la collision entre le missile et le vaisseau nous avons cherché chaque de notre côté car nous sommes rendus compte que c'était assez complexe et c'était plus simple de trouver la solution toute seule. Ensuite nous avons mis en commun. J'ai essayé de faire la page web mais Chloé l'a fait et m'a demandé mon avis.

Malgré le temps passé à rester bloqué sur les deux collisions entre le missile et le vaisseau et entre la punaise et le missile, nous n'avons pas eu assez de temps pour régler les derniers petits détails. Par exemple faire descendre des vies ou bien encore la page d'accueil. Nous ne pensions pas que les collisions étaient complexes.

IV. Problème rencontré

Au tout début, nous avons eu un problème pour le fond d'écran, nous n'arrivions pas à trouver les longueurs de l'image pour pouvoir mettre les coordonnées du centre de l'image alors nous testions des coordonnées et à chaque fois il manquait des parties de l'image. Au final, nous avons trouvé les coordonnées dans les propriétés de l'image.



Ensuite nous avons des problèmes de collision entre missile et vaisseau et entre punaise et. J'avais trouvé une solution pour le vaisseau et le missile avec un professeur. Chloé avait trouvé une autre solution plus complexe pour la collision entre missile et punaise. Nous avons combiné les deux et les missiles n'explosaient pas quand la punaise touchait le missile. Nous avons eu aussi un problème avec la vitesse de la punaise qui accélère au fil du temps. Au final nous avons réglé tous ces problèmes et le jeu fonctionne.

V. Programmation

Nous avons 5 fonctions dans la boucle principale. Ces fonctions sont `afficher_vies()` qui permet de définir le nombre de vies, `déplacer(event)` qui permet de déplacer le vaisseau avec les touches, `créer_missile()` qui permet de choisir aléatoirement les coordonnées d'un missile, `déplacer_missile()` qui permet de descendre les missiles et de faire la collision entre le missile et le vaisseau, `créer_punaise()` qui permet de lancer les punaises vers le haut à partir des coordonnées du vaisseau et la dernière fonction, `déplacer_punaise()` qui permet de gérer les collisions entre punaise et missile et d'effacer les missiles lorsqu'il y a collision. Les trois dernières fonctions s'arrêtent lorsqu'il n'y a plus de vies.

```
def afficher_vies():
    global nombre_vies
    Vies=Label(text="Nombre de vies restantes :"+str(nombre_vies))
    Vies.place(x=25,y=2.5)
    Space_Game.after(10,afficher_vies)

def déplacer(event): ##Déplacer le vaisseau
    global x_vaisseau,vitesse_déplacement_vaisseau
    if event.keysym=="Right":
        jeu.move(vaisseau_aff,vitesse_déplacement_vaisseau,0) ##Déplace le vaisseau à droite
        x_vaisseau+=15
    if event.keysym=="Left":
        jeu.move(vaisseau_aff,-vitesse_déplacement_vaisseau,0) ##Déplace le vaisseau à gauche
        x_vaisseau-=15

jeu.bind_all("<Key>",déplacer) ##Associe la touche à la fonction déplacer

def créer_missile():
    global x_missile,id_missile,y_missile,etat_missile,vitesse_création_missile, nombre_vies
    x_missile_init=randint(0,990) ##Choisit une valeur en x aléatoire
    x_missile.append(x_missile_init) ##Cette valeur est donc immédiatement envoyée dans la liste stockant les coordonnées de la météorite
    img1=jeu.create_image(x_missile_init,50,image=missile) ##Création d'un missile avec les coordonnées en x définies au dessus
    id_missile.append(img1) ## Cette météorite est alors référencée dans une liste
    y_missile.append(y_missile_init) ##La coordonnée en y de la météorite est alors elle aussi référencée dans une liste pour la connaître en temps réel
    etat_missile.append("NORMAL") ##On choisit de mettre un état à la météorite qui permettra de détecter ou non une collision
    if nombre_vies==0:
        return ##Stoppe la fonction lorsqu'il n'y a plus de vie
    Space_Game.after(vitesse_création_missile,créer_missile) ## Répétition de cette fonction toutes les "vitesse_création_missile" ms

def deplacer_missile():
    global y_missile_init,x_vaisseau,x_missile,y_vaisseau,nombre_vies,vaisseau_aff,id_missile,id_punaise
    for i in range(0,len(id_missile)):
        if y_missile[i]<732:
            jeu.move(id_missile[i],0,vitesse_max_missile)
            y_missile[i]+=vitesse_max_missile
            if y_vaisseau+150>y_missile[i]>y_vaisseau and x_vaisseau-30<x_missile[i]<x_vaisseau+30:
                if etat_missile[i]=="NORMAL":
                    etat_missile[i]="HIDDEN"
                    jeu.delete(vaisseau_aff)
                    nombre_vies-=1
                if nombre_vies==0:
                    Space_Game.after(500)
                    vaisseau_aff=jeu.create_image(x_vaisseau,y_vaisseau+64,image=vaisseau)
                if nombre_vies==0:
                    return ##Stoppe la fonction lorsqu'il n'y a plus de vie
    Space_Game.after(30,deplacer_missile)
```

VI. Le site Web

Pour le site web nous avons décidé de faire une page simple avec la règle du jeu et la présentation. Nous avons utilisé le langage html et css. Il est possible de télécharger le jeu sur notre site.

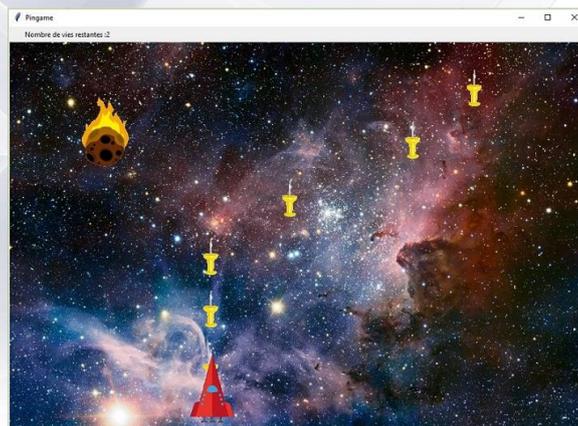
```
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" />
    <title>Le Pingame</title>
  </head>

  <body>

    <p>Site créé par Aurélie BARRÉ et Chloé GEOFFROY</p>
    <h1><center>Le Pingame</h1>
    <p><center></center><span><span></p>
    <h2>Qui sommes-nous ?</h2>
    <p><span><span>Nous sommes deux élèves de Terminale S-SI spécialité Informatique et Sciences du Numérique au lycée Jean-Auguste Margueritte de Verdun.<span><span></p>
    <h2>Dictaciel</h2>
    <p><span><span>Utilisez les flèches gauche et droite du clavier pour déplacer votre vaisseau et envoyer des punaises pour détruire les missiles qui vous tombent dessus.<span><span></p>
    <p><span><center></center></p>
    <h2>Vous voulez le télécharger ?</h2>
    <p><span><a href="Pingame.rar" > Cliquez ici pour le télécharger ! </a>

  </body>
</html>
```

Le Pingame



Qui sommes-nous ?

Nous sommes deux élèves de Terminale S-SI spécialité Informatique et Sciences du Numérique au lycée Jean-Auguste Margueritte de Verdun.

Dictaciel

Utilisez les flèches gauche et droite du clavier pour déplacer votre vaisseau et envoyer des punaises pour détruire les missiles qui vous tombent dessus. Vous avez 3 vies. Dès lors que vous n'en n'avez plus, le jeu s'arrêtera. Bon courage et bon jeu !

Conclusion

Ce jeu nous a permis de découvrir de nouveaux outils sur python et de chercher des solutions. Pingame nous a aussi permis de surmonter les problèmes rencontrés. Même si il reste encore des petits réglages, notre jeu fonctionne.