

Compte-Rendu : Street Fighter (en moins bien) – Elias

La première difficulté dans la réalisation de la découverte de la bibliothèque tkinter. nécessité deux ou trois semaines avant le commencement de la programmation dite et, au cours même de la programmation, certains passages ont dû plusieurs fois afin d'aboutir à un fonctionnant.

C'est notamment le cas de la fonction qui personnage de sauter (cicontre).

```
def descente1 ():  
    if terrain.coords(perso1)[1]<300 :  
        terrain.move(perso1,0,10)  
        terrain.after(30,descente1) def  
montee1 () :  
    if terrain.coords(perso1)[1]>150 :  
        terrain.move(perso1,0,-10)  
        terrain.after(30,montee1)  
    elif terrain.coords(perso1)[1]<300 :  
        descente1()
```

ce projet a été
Cela a

proprement

être revus
programme

permet au

```
perso1=terrain.create_image(250,300,anchor=NW,image=im1)
```

Plus tôt dans la programmation, un autre problème a été de faire apparaître les personnages sans fond derrière eux. La solution était d'inclure les

images dans un canevas et de ne pas en faire, comme essayé dans les premières tentatives, des widgets à part (le résultat final ci-contre).

A vrai dire, la partie du travail qui a dû prendre le plus de temps est la création des sprites (même si celle-ci s'est faite sans aucun talent artistique...). 12 personnages sont jouables. Ce travail fastidieux sur le plan graphique a été accompagné d'une part de documentation sur les pays choisis (ces pays ont été pris pour être en contraste avec les traditionnels du jeu original) pour rédiger les courtes présentations des personnages et, dans le cas de Paenui et Pety par exemple, trouver des noms qui correspondent bien à ce que l'on peut trouver dans la réalité dans ces pays.



Un des problèmes qui subsistent concerne le saut. En effet, le joueur doit appuyer sur la touche et relâcher immédiatement, sous peine de voir son personnage « trembler », ce qui rend la partie impossible car il est en constant saut et que les deux personnages doivent être à la même hauteur pour que les attaques fonctionnent (c'est pour ça d'ailleurs que le saut peut être utile en partie).

```
if x2-x1<50 :
```

```
    terrain.itemconfig(barredevie2,fill="black")
```

```
elif x2-x1<100 :
```

```
    terrain.itemconfig(barredevie2,fill="red")
```

```
elif x2-x1<200 :
```

```
    terrain.itemconfig(barredevie2,fill="yellow")
```

```
else :
```

```
    terrain.itemconfig(barredevie2,fill="green")
```

Si l'utilisation de turtle était auparavant envisagée, tkinter a finalement été la seule bibliothèque utile (et random pour avoir un personnage au hasard si aucun n'a été sélectionné, alors qu'avant l'ajout de choice au programme c'était Omega Z par défaut). La barre de vie a ainsi été conçue de façon à changer de couleur en fonction de la portion de vie restante.

Pour les attaques et les sauts, il a été envisagé d'utiliser `time.sleep()`, mais, cela ne fonctionnant pas, une solution plus efficace a été trouvée : `.after`.

A propos des attaques justement, des attaques spéciales (attaque, soin ou téléportation, voire mélange de deux de

ces capacités), utilisables trois fois dans la partie, ont été ajoutées. Celle du personnage mystérieux (Omega Z) est bien censée empêcher l'adversaire d'utiliser sa propre attaque spéciale.

Les longueurs ont été réglées sans logique particulière, convenir à l'écran utilisé pour concevoir le jeu. Aussi risque qu'elles ne conviennent pas tout à fait à certains

```
jeu.geometry("1500x600+0+33")
```

mais pour
existe-t-il le
ordinateurs.

En réalité, la plus grande difficulté n'a pas tant été de l'ordre de la programmation que sur la « métaprogrammation » ; l'ordinateur utilisé a eu du mal à fonctionner, ce qui a forcé à reprendre une version enregistrée antérieurement et donc moins achevée. Ce retard a plus ou moins été rattrapé néanmoins.

Globalement, finalement tout s'est pas trop mal passé...

PS : Il y a deux cheat-codes (PYRAX et MUNJA), qui permettent à l'utilisateur concerné (le premier pour le joueur 1, le deuxième pour le joueur 2) de gagner la partie dans la majorité des configurations (à moins de n'avoir 5 fois moins de PV que l'adversaire, auquel cas l'attaque spéciale de Ahuitzotl serait bien utile...).

Novum PS : Il y a eu des ajouts depuis la rédaction du compte-rendu :

- Les arènes (un personnage attaque plus fort s'il est du même pays que l'arène et si l'adversaire n'est pas du même pays).
- Les fonds événementiels (si aucune arène n'est sélectionnée, une arène par défaut dépendant de la date dans l'année est définie, la date de changement étant le 15 de chaque mois ; par exemple, entre le 15 janvier inclus et le 15 février, il y a un fond spécial « nouvel an chinois »).
- Le mode « contre l'IA » (le J2 est en mode automatique et affronte le joueur « organique », le J1, qui doit donc gagner contre un personnage, tout en étant seul).
- Le mode « 2v2 » (4 joueurs s'affrontent, avec $J1=J2.2$ et $J2=J1.2$, mais, contrairement à ce que le nom semble indiquer, les joueurs ne sont pas forcément en équipes de 2, c'est « chacun pour soi » et à la fin l'un des 4 joueurs gagne)