

Projet personnel :

Jeu de rôle basé sur l'univers du seigneur des anneaux (sur python) :

implémentation :

- les personnages sont décrits à l'aide de listes qui contiennent différents statistiques (résistance, portée, points d'attaque, agilité etc...)
- des fonctions permettent les différentes actions : duel prenant en argument deux listes représentant des personnages et confrontant leurs statistiques pour en déterminer le vainqueur, événements variés (découvertes, rencontres, dialogues etc...)
- des graphes qui représentent les cartes (les sommets du graphe sont colorés en fonction de ce qu'ils représentent : villes, lacs, chaîne de montagnes, forêts ...) la nature de ces sommets constitue un des arguments de fonctions précédemment citées (le duel dans la forêt avantage les elfes, par exemple).
- On peut ainsi former des « intrigues » avec des fonctions qui s'appellent mutuellement selon les choix des joueurs (python présente la fonctionnalité de prendre en compte les demandes des joueurs dans la console, et le programme peut ainsi réagir).

Exemple : (les propriétés décrites au début sont les éléments des listes « ci » représentant les caractères.

```
1 # vie
2 # armure
3 # magie (1 ou 0)
4 # puissance(-1 si magie = 0)
5 # physique
6 # vitesse (agilité)
7 # race (humain, nain, elfe, orque, magicien, goblin, spectre, ent, dragon)
8
9
10 def terrain(c,milieu):
11     d=c
12     if "humain" in c[6] == True:
13         if milieu == "plaine":
14             d[5] += 0.2
15         elif milieu == "forêt" or milieu == "montagne":
16             d[5] -= 0.2
17     elif "elfe" in c[6] == True:
18         if milieu == "forêt":
19             d[5] += 0.25
20         elif milieu == "montagne":
21             d[5] -= 0.25
22
23 def duel(c1,c2):
24     v1 = c1[0]
25     v2 = c2[0]
26     if c1[2] == 0:
27         if c2[2] == 1:
28             v1 -= c2[3]
29     else:
30         v2 -= c1[3]
31         if c2[2] == 1:
32             v2 -= c1[3]
33     ratio = c1[5]/c2[5]
34     while v1 > 0 or v2 > 0:
35         v1 -= (1/c1[1]) * c2[4] * (1/ratio)
36         v2 -= (c/c2[1]) * c1[4] * ratio
37     if v1 > 0:
38         return c1
39     elif v2 > 0:
40         return c2
41     else:
42         return []
43
44
45 def duel_milieu(c1,c2,milieu):
46     d2 = terrain(c2,milieu)
47     d1 = terrain(c1,milieu)
48     return duel(d1,d2)
```