

Rapport Base de données réparties

TP2

Achraf Alloubi - ALLA11069909
Laila El Khatibi El Hanana - ELKL23599905
Riwan Le Gal - LEGR27039804
Léo Mérel - MERL27109808

Exercice 1: La guérison

Le but de cet exercice était de récupérer les sorts de chaque monstre puis d'inverser les données pour avoir tous les monstres disposants du sort recherché.

On a donc d'abord effectué un crawling sur le site référençant tous les monstres pour récupérer tous les sorts dont ils disposent. Il a fallu être minutieux pour ne pas louper d'informations sur les sorts qui peuvent être à plusieurs endroits de la page de chaque monstre. On a ensuite stocké tous les résultats dans un json au format suivant :

```
"name": "Aasimar",  
"spells": [  
  "daylight",  
  "bless",  
  "command",  
  "protection from evil",  
  "detect magic",  
  "guidance",  
  "stabilize"  
]
```

Après un traitement sur ces données, on les a donc transformées pour qu'elles soient au format **spell** → [**monstre 1, monstre2 ...**] au lieu de monstre → [spell1, spell2 ...].

Pour cela, on va utiliser un qui va nous permettre d'appliquer une certaine fonction fun1() aux différents éléments de notre RDD.

```
#Fonction pour la création de tuples (sort, nom créature)
def fun1(x):
    nomcreature=x[0]
    sorts = x[1]
    tuples = []
    for sort in sorts:
        tuples.append((_sort.strip(),nomcreature.strip()))
    return tuples

output = monster.rdd.flatMap(lambda x: fun1(x))\
    .groupByKey()
```

La fonction fun1() nous permet d'obtenir la liste de tuples avec comme premier élément le nom du sort (spell) et comme second élément le nom de la créature.

On finit par utiliser la méthode groupeByKey qui va nous permettre de regrouper les tuples en fonction du nom du sort.

Finalement, on a enregistré le tout dans un json au format suivant:

```
"name": "Acid Spit",
"monsters": [
    "Azruverda",
    "Chon Chon",
    "Chon Chon Swarm",
    "Hive Warrior",
    "Thessalhydra"
]
```

Exercice 2 (Choix #1)

L'objectif de cet exercice était de créer un outil de recherche de sorts. Pour cela, on a récupéré notre fichier json obtenu à la suite d'un crawling dans le TP1, dans lequel on récupérait toutes les informations liées à chacun des sorts : son nom, les classes pouvant l'utiliser (et à partir de quel niveau), ses composantes (V, S, M notamment) ...

Ce fichier json nous a servi à créer une RDD sur laquelle on a pu faire nos requêtes par la suite, pour trouver les sorts ayant telle ou telle caractéristique recherchée.

On a d'abord créé une interface graphique simple sur Tkinter (python) qui permet à l'utilisateur de sélectionner les caractéristiques qu'il souhaite pour le sort qu'il recherche, à l'aide de champs de texte, boutons, menu déroulant ...

Pour sa recherche l'utilisateur peut renseigner le niveau du sort (min et/ou max), les classes qui peuvent utiliser le sort. Il est possible de spécifier le nom du sort ou bien une chaîne de caractères incluses dans le nom du sort. Enfin, une sélection sur le lancement de sort V,S,M peut être paramétrée. Tous ces éléments constituent l'ensemble des critères paramétrables par l'utilisateur.

Une fois que l'utilisateur a sélectionné tout ce qu'il souhaite, il peut cliquer sur le bouton *submit*. Dans le code, c'est à ce moment-là qu'on vient récupérer les valeurs de tous les composants de l'interface pour les convertir et faire une requête SQL.

Voici un exemple de requête SQL sur notre JSON :

```
SELECT name FROM spells WHERE (spells.bloodrager >= 3 AND spells.bloodrager <= 4 AND spells.magus >= 3 AND spells.magus <= 4 AND spells.witch >= 3 AND spells.witch <= 4) AND (spells.Components LIKE '%V,%') AND (spells.Components LIKE '%S,%')
```

Après avoir effectué une requête, nous obtenons une liste de sorts répondant aux critères spécifiés. Cette liste est affichée dans notre interface au travers d'une listbox (**Results**). Il est ensuite possible de sélectionner un sort de la liste et d'afficher l'image correspondante en appuyant sur le bouton "show Image".

Exemple :

Exemple de recherche par le nom du sort :

Spell search

Spell level min: max:

Classes

adept
alchemist
antipaladin
arcanist
bard
bloodrager
cleric
druid

Or And

Spell name

Begins with Ends with Anywhere

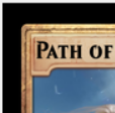
VSM

☐ ☒ ☐ ☐ ☐

Results

Find the Path
Nature's Paths
Path of Glory
Path of Glory, Greater
Path of the Winds

Show Image



Submit