

Rapport de Projet MAC : **TrivialBot**

Cahiers des charges :

1. MVP(Most Viable Product)

Les points suivants sont requis pour que notre produit soit fonctionnel :

1. L'utilisateur peut commencer une partie
2. L'utilisateur peut changer les paramètres de la partie
3. Le bot possède un LeaderBoard pour garder une trace des meilleurs joueurs
4. Les utilisateurs peuvent ajouter des questions à la base de donnée
5. Différentes statistiques sont stockées dans la base de donnée

2. Fonctionnalités Supplémentaires

1. Utilisateurs peuvent Upvote ou DownVote des questions
2. Implémentation de divers Bonus

3. Fonctionnalités implémentées

- L'utilisateur peut lancer une partie avec **/start**
- L'utilisateur peut stopper une partie avec **/stop**
- L'utilisateur peut répondre à des questions à choix multiple
- Les questions sont tirées de l'API OpenTriviaDatabase
- Lorsque qu'un nouveau joueur rejoint une partie il est stocké dans une Graph Database
- L'utilisateur peut voir les scores de tous les participants de la partie en cours
-

Modèles de Données :

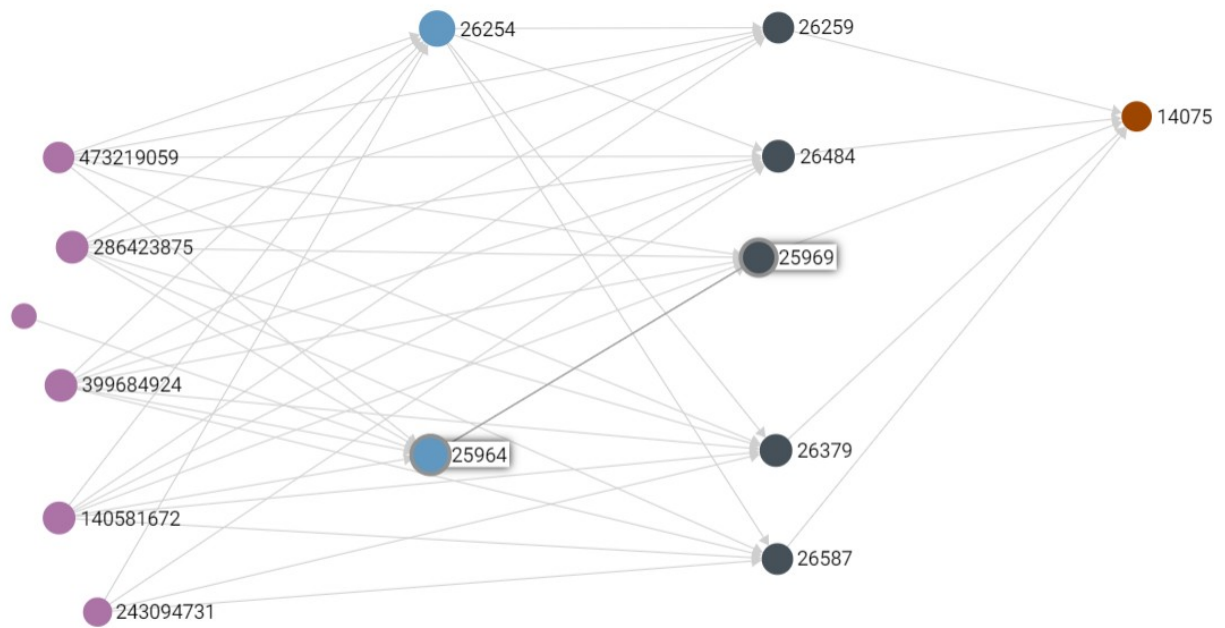
1. Modèle de données documents

Pour le modèle documents nous utilisons Open Trivia database qui nous fournis les questions et les réponses sous la forme d'un json :

```
{
  "response_code":0,
  "results":[
    {
      "category":"Animals",
      "type":"multiple",
      "difficulty":"easy",
      "question":"What is the fastest land animal?",
      "correct_answer":"Cheetah",
      "incorrect_answers":[
        "Lion",
        "Thomson's Gazelle",
        "Pronghorn Antelope"
      ]
    }
  ]
}
```

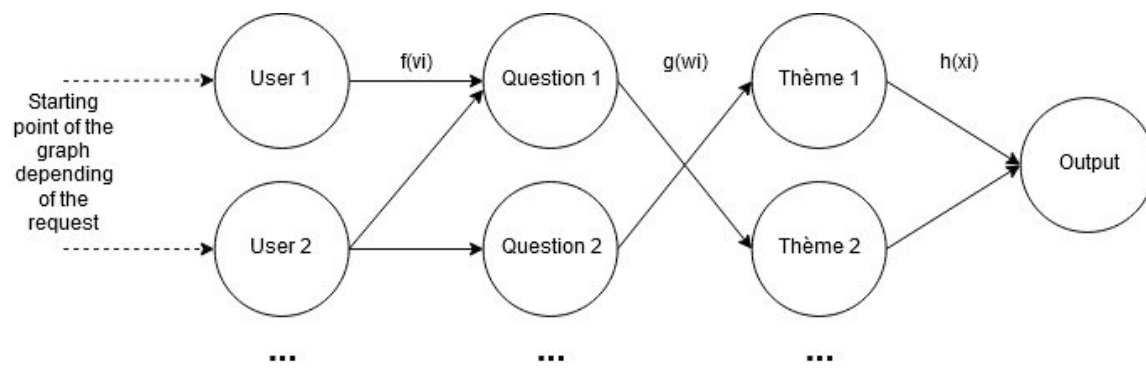
L'avantage de cette base de donnée est qu'elle possède une API python permettant de facilement l'intégrer au code de notre Bot.

2. Modèle de données Graphe



violet : utilisateurs
bleu : parties
noirs : questions
marron : catégorie

Explication des requêtes avancées :



v = Winrate of the user for the linked question

f = Function used to transform the winrate to weigh the edge, so that the weight is low when the winrate is high.

w = Average winrate of the question

g = Function used to weigh the edges, so that when the average winrate is low, the weigh is high.

x = Like/Dislike ratio of the teme

h = Function used to weigh the edges, so that when the dislike ratio is high, the weigh is low.

Goal: This graph is designed to perform a graph request. When a user perform a request, the starting node is set to his node. We then perform a shortest path algorithm. The output is the recommended theme for the user.

Structure du code :

Notre code est structuré selon 3 packages principaux :

- Le bot
- Les bases de donnés
- Les modèles de données

La classe principale du bot est TriviaBot, cest dans celle-ci que les commandes Telegram sont interprétées.