**METHODE D’ACCES AU DONNEES**

**Rapport**

**Projet**

Christen, Dupraz & Mottier

Table des matières

[1. Description 3](#_Toc61814081)

[2. Fonctionnalités 3](#_Toc61814082)

[3. Modélisation des données 4](#_Toc61814083)

[Origine des données 5](#_Toc61814084)

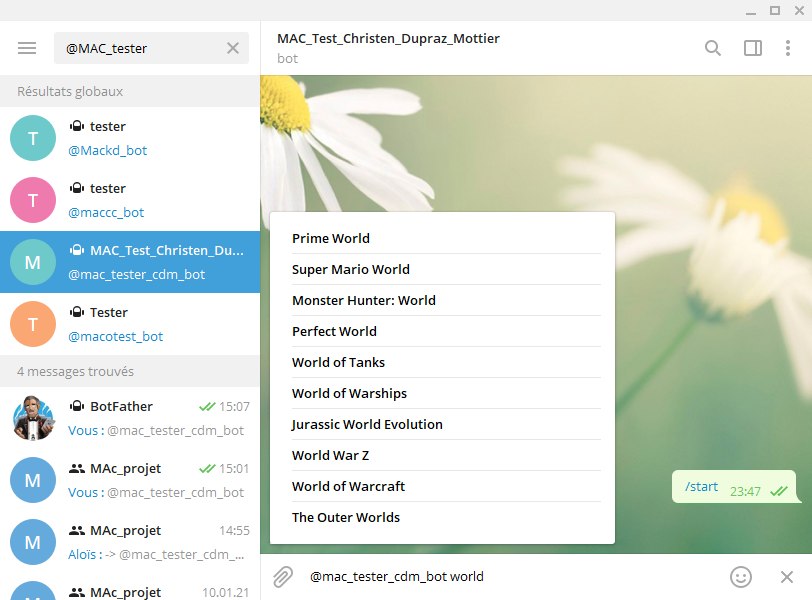
# Description

Le but du projet est de créer un bot sur le client de messagerie Telegram. Ce bot aura pour but de pouvoir interagir avec la plateforme de streaming Twitch (<https://dev.twitch.tv/docs/api/>) qui dispose d’une API.

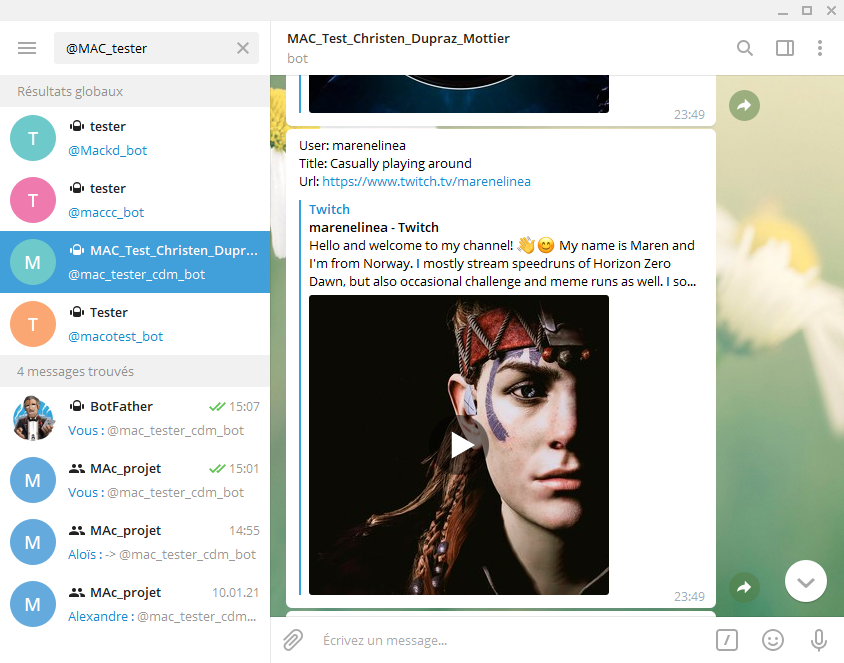
Le bot aura plusieurs fonctionnalités qui seront énoncées au point suivant.

# Fonctionnalités

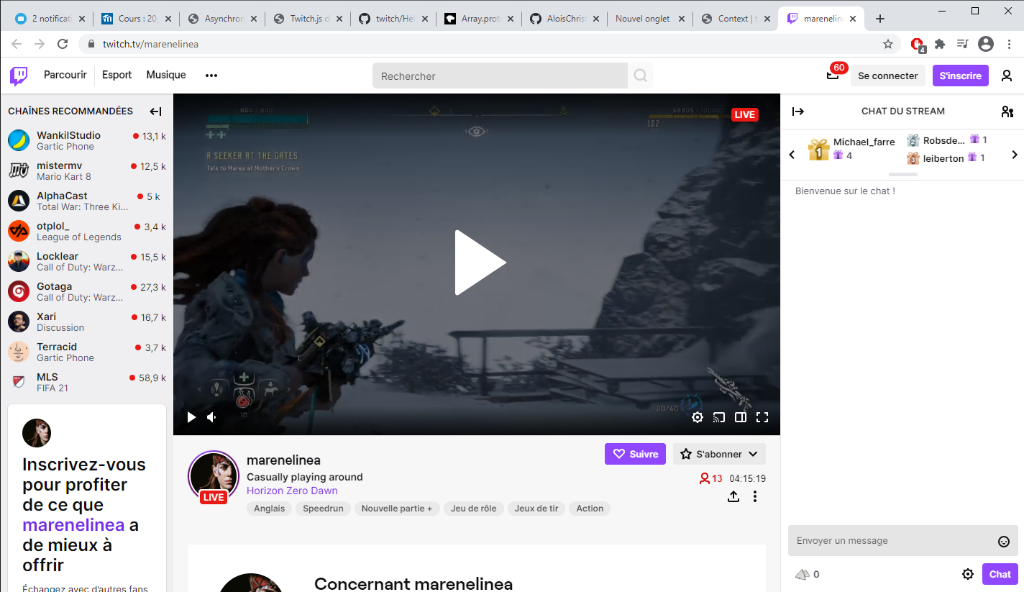
Le bot vous propose de liker les jeux disponibles dans le top 500 des jeux streamés sur Twitch. L’image ci-dessous montre un exemple de recherche de jeu et l’affichage instantané des résultats.



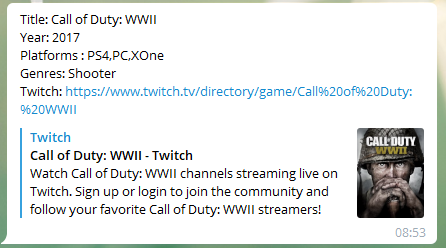
En entrant la commande « /recommendstreamer », le bot affiche une liste de streamers qui pourraient vous plaire, à la condition que vous ayez liké assez de jeux. La capture ci-dessous montre une proposition de stream d’Horizon Zero Dawn



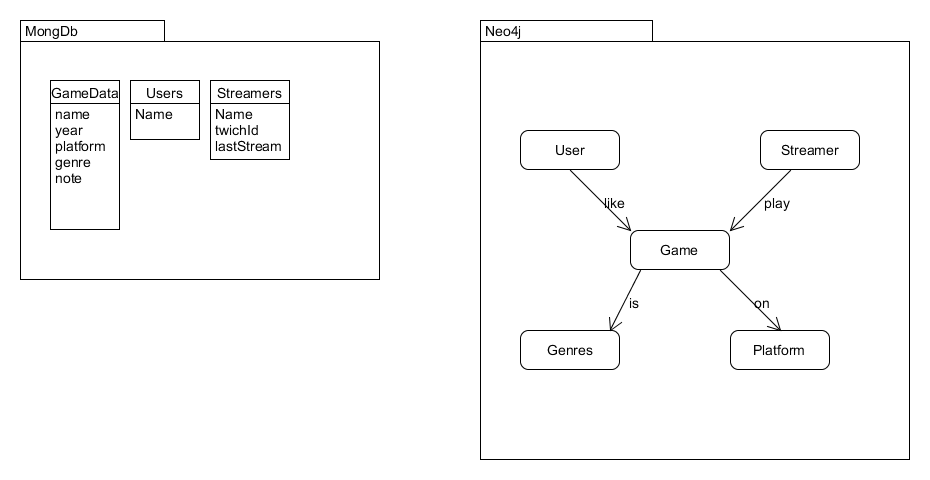
La sélection d’un streamer vous proposera d’aller voir son stream, comme ci-dessous avec la proposition de stream précédente.



En entrant la commande « /recommendstreamer, le bot affiche une liste de jeux qui pourraient vous plaire, à la condition que vous ayez liké assez de jeux.



# Modélisation des données



Dans notre base MongoDB, nous avons deux collections :

* Games : cette collection contient les top jeux streamés sur Twitch. On peut y retrouver le nom du jeu, sa date de sortie, les plateformes sur lesquelles le jeu peut être joué, les genres qui lui sont attribués et son id Twitch.
* Streamers, le top 3 des streamers pour la première partie des jeux, et le top 1 des streamers sur la seconde partie des jeux. La collection est composée de l’id Twitch et du nom streamer et son basename (nom dans l’url).

Dans notre base Neo4j, nous avons 5 types de nodes :

* User
* Streamer
* Game
* Genre
* Platform

Et contient aussi 4 types de relations :

* LIKE : User → Game
* PLAYS\_TO: Streamer → Game
* BELONGS\_TO: Game → Genre
* PLAYED\_ON: Game → Platform

## Provenance des données

Les données insérées dans MangoDB pour les jeux proviennent de [Kaggle](https://www.kaggle.com/) et les données sur les streamers viennent directement de Twitch en passant par leur [API](https://dev.twitch.tv/docs/api/).

## Intégration des données

Nous intégrons les données des jeux sur MangoDB grâce au package « mangodb » pour NodeJS. Nous avons modifié nos données de base pour n’avoir que le top 500 des jeux streamés sur Twitch.

Pour intégrer les données des streamers, nous créons un CSV grâce à l’API de Twitch. Nous insérons seulement des streamers qui jouent à un jeu qui est déjà dans notre base.

# Explication des requêtes

# Améliorations possibles

En l’état, notre bot pourrait être amélioré de nombreuses façons. Tout d’abord, plus de jeux pourraient être stockés dans la base de données, sans se limitant aux 500 top jeux de Twitch. Il serait aussi possible de laisser l’utilisateur rentrer lui-même des jeux, ou des streamers (par exemple s’il se lance dans cette activité).

Une autre amélioration évidente serait de pouvoir liker également des streamers, et que la recommandation de streamers prenne en compte ces préférences. Ces mêmes recommandations pourraient de plus intégrer d’autres paramètres : année de sortie, développeur, langue du stream, etc.