

Documentation du Projet — OcciLan Stats

Last updated: aujourd'hui

1. Présentation générale

Contexte

OcciLan Stats est un projet interne destiné à centraliser et visualiser les statistiques du tournoi **OcciLan**, un événement étudiant organisé chaque année. Il s'adresse avant tout aux **joueurs** et **organiseurs** de l'OcciLan, afin de suivre les performances, les résultats et les statistiques de chaque équipe et joueur pendant et après le tournoi.

Objectif principal

Automatiser la récupération et l'analyse des statistiques des matchs de la LAN (issues de Riot Games), à partir de simples liens OP.GG ou d'un export du tournoi Toornament. Le système génère ensuite : - des statistiques individuelles et collectives (KDA, Winrate, CS/min, Vision/min, etc.), - des tableaux de performances par équipe et joueur, - des visualisations interactives accessibles via une interface Streamlit.

Public cible

- Les **joueurs** de l'OcciLan (pour consulter leurs propres stats ou comparer leurs performances), - Les **organiseurs** du tournoi (pour suivre les résultats, communiquer, archiver les données des éditions précédentes).

Périmètre

Le projet est focalisé sur **League of Legends**, mode **Draft Tournoi 5v5**, pour un tournoi de 16 équipes et environ 100 matchs par édition. Il ne vise pas le niveau professionnel, mais un cadre amateur et étudiant.

2. Description fonctionnelle

Cas d'usage principaux

1. **Organisateur** : prépare une édition, importe la liste d'équipes depuis Toornament (ou un CSV), puis génère automatiquement les statistiques globales.
2. **Joueur** : consulte les statistiques détaillées de son équipe et de ses matchs.

Scénario d'utilisation

1. L'organisateur choisit une édition (ex. « OcciLan #7 »).
2. L'application récupère la liste des 16 équipes et leurs joueurs via Toornament ou un CSV.
3. Chaque lien multi-OP.GG est analysé pour extraire les Riot IDs des joueurs.
4. Les joueurs sont identifiés dans l'API Riot (PUUID) et leurs matchs récents sont collectés.
5. Les matchs correspondant à la période du tournoi sont filtrés, normalisés et analysés.
6. Les résultats et statistiques s'affichent automatiquement sur le tableau de bord.

Fonctionnalités principales

- **Vue d'ensemble** : répartition par Elo, nombre total de matchs, durée moyenne, winrate global.
- **Équipes** : statistiques agrégées par équipe (victoires, KDA, rôles, performances par match).
- **Joueurs** : profil complet avec KDA, KP, CS/min, Vision/min, winrate, et champions joués.
- **Matches** : historique détaillé avec résultats, durées, compositions, et performances individuelles.
- **Administration** : rafraîchissement du cache, ajout d'une nouvelle édition, export des données.

3. Données et sources

Structure du tournoi OcciLan

Chaque édition comprend : - **16 équipes** de 5 joueurs (80 joueurs au total), - **5 matchs de poule** (système suisse), - une **phase éliminatoire** à double élimination (BO3 et BO5), - environ **88 à 100 matchs** au total par édition.

Origines des données

- **Toornament API** : permet d'obtenir la liste des équipes et participants.
- **CSV manuel** : alternative simple contenant `[team_name, multi_opgg]`.
- **OP.GG** : uniquement utilisé pour extraire les Riot IDs des joueurs.
- **Riot API** : principale source de données pour les matchs, statistiques et profils (Account, Match V5, Summoner V4).

Types de données manipulées

Type	Description	Exemple
Équipe	Nom, région, joueurs, résultats	KCDQ, EUW
Joueur	Riot ID, PUUID, rôle, Elo	PlayerA#EUW, Top
Match	Durée, vainqueur, score, champion, KDA	27:43, Win, 5/2/9
Champion	Nom, nombre de matchs, taux de victoire	Aatrox, 8 parties, 62.5% WR

4. Processus global (Dataflow)

1. **Entrée** : Toornament API ou CSV → liste d'équipes + liens multi-OP.GG.
2. **Extraction** : parsing des URLs multi pour récupérer les Riot IDs.
3. **Résolution** : appel à l'API Riot → obtention des PUUID.
4. **Collecte** : récupération des matchs récents (Match V5) pour chaque joueur.
5. **Filtrage** : conservation uniquement des matchs de la période de la LAN.
6. **Analyse** : calcul des indicateurs (KDA, KP, CS/min, Vision/min, etc.).
7. **Agrégation** : regroupement des stats par joueur, équipe et champion.
8. **Visualisation** : affichage dans Streamlit.

5. Schéma des données

Table	Description	Clé primaire
edition	Informations sur le tournoi (dates, nom, format)	edition_id
team	Équipe (nom, région, membres)	team_id
player	Joueur (PUUID, Riot ID, équipe, rôle)	puuid
match	Partie (durée, date, patch, queue)	match_id
participant	Données individuelles par match	match_id + puuid
player_stats	Moyennes par joueur	edition_id + puuid
player_champ_stats	Moyennes par champion	edition_id + puuid + champion
team_stats	Statistiques par équipe	edition_id + team_id

6. Architecture technique

Technologies utilisées

- **Python** pour le traitement et les appels API.
- **Streamlit** pour l'interface graphique.
- **Pandas** pour la manipulation de données.
- **DuckDB / SQLite** pour le stockage local et la mise en cache.
- **Riot API** pour les données de match.

Diagramme simplifié

```
[Toornament / CSV] → [Parser OP.GG] → [Riot API]
                        ↓
                    [Nettoyage + KPIs]
                        ↓
                [Cache DuckDB / Streamlit]
                        ↓
                [UI : Stats & Graphiques]
```

7. Sécurité et conformité

- La clé Riot API est privée et stockée côté serveur.
- Respect des CGU de Riot Games et d'OP.GG.
- Pas de données personnelles sensibles.
- Respect RGPD : pseudonymisation, usage interne uniquement.
- Accès restreint à la console d'administration.

8. Limites et risques

Type	Description	Niveau
Technique	Quotas d'appels Riot API (20 req/s)	Moyen
Données	Données de matchs customs parfois incomplètes	Faible
Performance	Chargement initial plus long (~2-3 min)	Moyen
Maintenance	Changements d'API Riot	Moyen

9. Perspectives d'évolution

- Intégration complète Toornament (arbre de tournoi, scores, phases).
 - Gestion multi-édition (historique des LANs précédentes).
 - Analyse par phase (Swiss / Playoffs).
 - Export automatique des résultats.
 - Intégration Discord (résumés et highlights automatiques).
-

10. Collaboration et gouvernance

- **Responsable technique** : Alexandre (Colfeo)
 - **Contributeurs** : membres de l'équipe d'organisation de l'OcciLan
 - **Fréquence des éditions** : 1 à 2 LAN par an
 - **Données traitées** : 16 équipes, 80 joueurs, ~100 matchs/édition
 - **Livrables** : tableaux de stats, graphiques, exports CSV/PDF
-

11. Annexes

Exemple de CSV d'entrée

```
team_name,multi_opgg  
KCDQ,https://op.gg/multisearch/euw?summoners=PlayerA%23EUW,PlayerB%23EUW  
HADES,https://op.gg/multisearch/euw?summoners=Foo%23BAR,Baz%23EUW
```

Endpoints Riot utilisés

- `/riot/account/v1/accounts/by-riot-id/{gameName}/{tagLine}`
- `/lol/match/v5/matches/by-puuid/{puuid}/ids`
- `/lol/match/v5/matches/{matchId}`
- `/lol/summoner/v4/summoners/by-puuid/{puuid}`

Indicateurs calculés

Nom	Formule
KDA	(Kills + Assists) / Deaths
KP	(Kills + Assists) / Total Kills de l'équipe
CS/min	(CS total) / Durée (min)
Vision/min	Vision Score / Durée (min)
Winrate	Victoires / Parties

Conclusion

OcciLan Stats vise à fournir aux joueurs et organisateurs une vue claire et automatisée des performances pendant la LAN. Grâce à l'usage des API officielles et à une architecture modulaire, le système peut évoluer vers un outil de suivi complet des éditions futures de l'événement.