

Simulateur de Parties d'Échecs PGN

Auteurs: Mathis & Baptiste LE ROUX

Date: 07/05/2023

Contact: mathis.le_roux@ensta-bretagne.org, bapt.leroux29@gmail.com

Table des matières

- [Introduction](#)
- [Utilisation](#)
- [Détail du Code](#)
- [Conclusion](#)

Introduction

Bienvenue dans le projet de simulateur de parties d'échecs PGN! Ce projet permet de lire un fichier PGN, de le traduire en une structure de données interne (AST), puis de simuler la partie d'échecs à l'aide d'une interface graphique basée sur Tkinter et de la bibliothèque Chess.

Utilisation

Pour utiliser ce projet, il est nécessaire de télécharger la librairie python-chess utilisée pour l'interface graphique.

Pour installer la librairie, réalisez cette commande sur windows:

```
pip install python-chess
```

Exécutez ensuite le script principal nommé "Analyse_partie.py" du projet avec un fichier .pgn en argument, comme suit:

```
python Analyse_partie.py exemple1.pgn
```

Pour faire plusieurs tests nous avons mis à votre disposition plusieurs .pgn, issues de parties jouées par des légendes du monde des échecs...

Vous pouvez bien entendu jouer vos propres parties en les ajoutant au dossier!

Le script principal du projet fera les opérations suivantes:

1. Lire le fichier PGN.
2. Effectuer une analyse lexicale et syntaxique du fichier PGN, créant un arbre de syntaxe abstraite (AST).
3. Visiter l'AST pour créer une représentation interne de la partie d'échecs.
4. Afficher la partie d'échecs sur une interface graphique, en permettant de naviguer à travers les coups.

Détail du Code

Le projet se compose de plusieurs composants principaux:

1. `pgn_lexer.py`: Ce fichier contient le code pour l'analyse lexicale du fichier PGN. Il transforme le texte du fichier en une série de tokens.
2. `pgn_parser.py`: Ce fichier contient le code pour l'analyse syntaxique des tokens, créant un AST.
3. `pgn_visitor.py`: Ce fichier contient le code pour visiter l'AST et créer une représentation interne de la partie d'échecs.
4. `pgn_interface.py`: Ce fichier contient le code pour l'interface graphique, qui affiche la partie d'échecs et permet de naviguer à travers les coups.
5. `utils.py`: Ce fichier contient des fonctions utilitaires pour convertir les mouvements dans le format approprié pour la bibliothèque Chess.

Conclusion

Ce projet a été une excellente occasion d'explorer le fonctionnement interne de l'analyse de texte et de la représentation de données complexes. En travaillant sur ce projet, nous avons pu approfondir notre compréhension de concepts tels que l'analyse lexicale et syntaxique, la création d'AST, la visite d'AST et la création d'interfaces graphiques en Python.

Le projet reste néanmoins assez simple, par manque de temps nous n'avons pas pu réaliser certains de nos objectifs, comme prendre en compte les commentaires et les afficher, la gestion d'erreur etc.

