Data Pipeline for Magic the Gathering Card Browser

John Tazioli

5 October 2022

Abstract

The goal of this project was to ingest Magic the Gathering card data into a database, extract the important pieces, and display on an easy to use and interactive application. The data is scraped from Scryfall. using the BeautifulSoup and requests library, stored in a Mongo database via Atlas, and interacted with through a streamlit app hosted on Streamlit Cloud.

Design

Magic the Gathering is a deck building game that started in 1993. There are over 70,000 cards with hundreds more published every year. A player uses between 40-100 cards in a game. It can quickly become overwhelming to sift through the options and build a successful deck. This project makes it easy to update the database with a python executable whenever a new deck is released. The database is stored for free on MongoDB Atlas, which hosts servers on Google Cloud. The code is stored on Github, which allows the application to be hosted by Streamlit Cloud. All of this equates to an easy to update and free solution.

Data

Scryfall maintains bulk downloads for all of their card objects. As of this paper there are 73,388 card objects, each with at least 64 features. The bulk download is a json file that is then stored in a mongo database called "MTG", in the "Cards" collection. The application only shows seven of the most relevant features in an interactive table.

Algorithms

Once every few months, the cards collection will need to be updated with new MTG card releases. This can be done with a single function in python. The streamlit application has the credentials stored in a secrets.toml file to query the MTG database as needed. To prevent the user from crashing the application with too big a query, the user is limited to viewing cards based on the set they released in. When the user selects a card from the interactive table, a find() query is submitted to the cards collection to return the URI of the image for that card so it can be displayed.

Tools

- Data Acquisition: requests and BeautifulSoup The URL for the bulk download changes every time there is an update. This required scraping the HTML to ensure the most updated bulk file is acquired.
- Cloud Storage: MongoDB Atlas A database service that operates through Google Cloud. Easy and streamlined process for accessing your database over the internet instead of localhost. Also, free for less than 512MB of storage.
- Database Interaction: PyMongo
- Web Applications: Streamlit and Streamlit Cloud Github integration and real time updates are valuable to the app development process.
- Report writing: LaTex

Communication

A five minute power point presentation and the code used will be hosted on my github.