**Intro**

The motivation of this project is to make a functioning card browser app for the card game Magic: The Gathering. The problem is that for the most part the apps that do exist while good, are quite bogged down with extra features as they all attempt to be a jack of all trades app. The solution is to have an application that’s simple and intuitive and allows for easy browsing of the cards in the game with filters and custom searches for ease of use.

**Motivation**

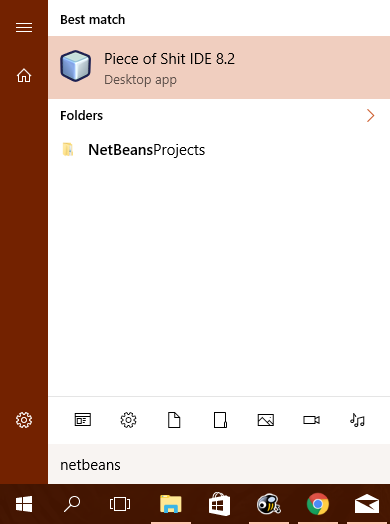
Magic the Gathering is an ever expanding card game and because of this it can be incredibly daunting to browse through all the cards since at this point there’s literally thousands of them that all do different things. This problem occurs pretty much whenever you want to build a new deck of cards for the majority of the formats that I play, as there are no restrictions as to what you can use so you have to filter through thousands of items to find just a few cards that you will be playing.

In all technicality the problem has already been solved and quite elegantly at that. There exists an online database of all the cards hosted by the creators of the game and there’s a multitude of applications that allow you to browse cards in the game as well as access other features for use in the game (collection management etc.). The most similar application to the one I propose is a program by the name of Decked, which is an all in one card browsing, deck building, card pricing and collection management app. I personally feel that decked is the pinnacle of what I could ever want for card management but I wouldn’t have a project if I just let the devs of that program have all the fun. My app would hypothetically be used by people who would want to look up cards on the fly, but again there already exists far more robust apps that do the same thing better.

**Project Details**

The majority of what I set out to accomplish was done. Searching and browsing cards works and quite well too. The only thing that was not accomplished was data visualization and some cordova specific uses as time constraints and formatting were a problem. For data visualization, the idea was to count the number of different card types distributed among the searched cards and display these values in a graph. So the problem with this was that to get your types the API returns them in an array, so in order to get that working I would have had to have a number of temporary ints to hold the different types (there’s 15 in total) which would be then used to count the different instances of each type and finally output it to the graph. Simply put by the time I figured out how to do it the project was already due, so it’s not getting implemented unless I feel like continuing this application on my spare time outside of it being an assignment. Overall I would say that I completed everything that the motivation required, and I’m honestly quite happy with this especially considering that a lot of other people did not finish their own projects.

The project environment used was Piece of Shit IDE 8.2 -- erm Netbeans 8.2 with Cordova installed to build for android.



(you thought I was joking)

Anyway, so I used PoS IDE 8.2 with cordova to code the actual application in HTML/Javascript/JQuery, etc. Standard web design stuff wrapped for android. For the user interface I used a UI Framework called Ratchet, which looks in my opinion quite good for mobile app development, and it helps that it’s recommended by Cordova’s website. In fairness the only reason I used Ratchet is because the thing’s incredibly quick to work with so I could just set it and forget it and the framework would do most of the work for me. I also used an external API called Magic the Gathering API. Effectively what this API does is return a JSON for me to parse based on whatever search querys I send it. So the API would do all the work for me and all I would have to do is send it what to return and format whatever it gave me into a list.

The hardware I worked with was obviously my laptop, and I tested the application by deploying it to my phone, which is a BLU Life One X running Android 5.1.1. This allowed me to test the application like an actual user since the android emulator we were supposed to use was a slow abomination and I simply did not have the patience to work with it when I could access things easier on my actual phone.

The best way to really show the features of the application would be to link to the structured demo I made for it, so that’s what we’re going to do: <https://www.youtube.com/watch?v=YiTM78kr7_Y> (Please forgive poor microphone quality)

The biggest problem I experienced, besides that array problem mentioned before was actually getting access to the API. It’s a server based API and the only javascript version of it available is an NPM plugin so since I’m not running a server sided program I couldn’t do that. So I spent like 4 hours trying to figure out how in the name of Shrek I was going to get access to the api until Julio mentioned offhandedly that I could just use a getJSON with a url constructed to search through the browser. That worked really nicely in the end.

If I’m going to be working on this in the future (which I just might) I think that the majority of things I’d be adding or changing would be for quality of life. For example I would move the different pages to separate files, add more search parameters and pass the search terms between pages more efficiently. Otherwise I can see myself eventually implementing a collection management function, and maybe a way to save things to a local directory.

**Conclusion**

Once again the game of Magic the Gathering is ever expanding and can be daunting to the players who may want to just see the different cards in the game so this application is designed in an attempt to capitalize on this by providing a simple streamlined way to browse and view cards on the fly. As mentioned before was the only person working on this project so I did everything myself.

**References**

<http://magicthegathering.io/> (Additional API used)

<http://goratchet.com/> (UI Framework)

<https://cordova.apache.org/> (code base used to wrap program for android)