Michigan State Park Vegetation Application

Executive Summary

Ethan Westerkamp

University of Wisconsin-Madison

Geography 778 Practicum

Background

The Michigan State Park Vegetation Application was created for the average recreation user to have simplistic access to an informative tool for discovering plant species existing in state parks in Michigan. As there is not total data on species information in state parks due to the scale of survey this would require, species can be estimated from the "Natural Communities" in the parks, defined by the Michigan Natural Features Inventory (MNFI) as "an assemblage of interacting plants, animals, and other organisms that repeatedly occurs under similar environmental conditions across the landscape". Complete information on these communities and the process of their definition can be found on the MNFI website: https://mnfi.anr.msu.edu/communities/classification . These communities have associated plant and animal species and so by examining the intersecting natural communities of state parks a robust list of species can be generated by a correctly formatted database. This web application demonstrates such a relational database and its data's query, display, and multi-directionality with the goal of allowing the user to easily search, filter, and read about specific plant species, be exposed to lesser-known species, and identify additional state parks which could contain a species.

Methods

This web application is intended to be similar to other web query apps in its interface outputting all information into a side panel and allowing the map to remain free of other information and maintain limited user interactions overall to allow for easy usage in retrieving desired information and avoiding more extraneous or technical details or features. All features within the database are structured to allow for updates to be pushed without disturbing all other layers. With the ArcGIS Maps SDK the intended features can easily be added and allow for simple modification and addition of any other components which could be possible in later development. Additional UI elements are created with HTML/CSS.

Structure

Software Diagram	
Mapping APIs	ArcGIS Maps SDK for JavaScript
Software Development	 (JavaScript, HTML)
Database	 (AGOL, ArcGIS PRO Geodatabase)
Hosting Environment	 (Github, AGOL)
Data Source	 (MNFI, Michigan DNR)

Figure 1: The Software Diagram of the application

Results

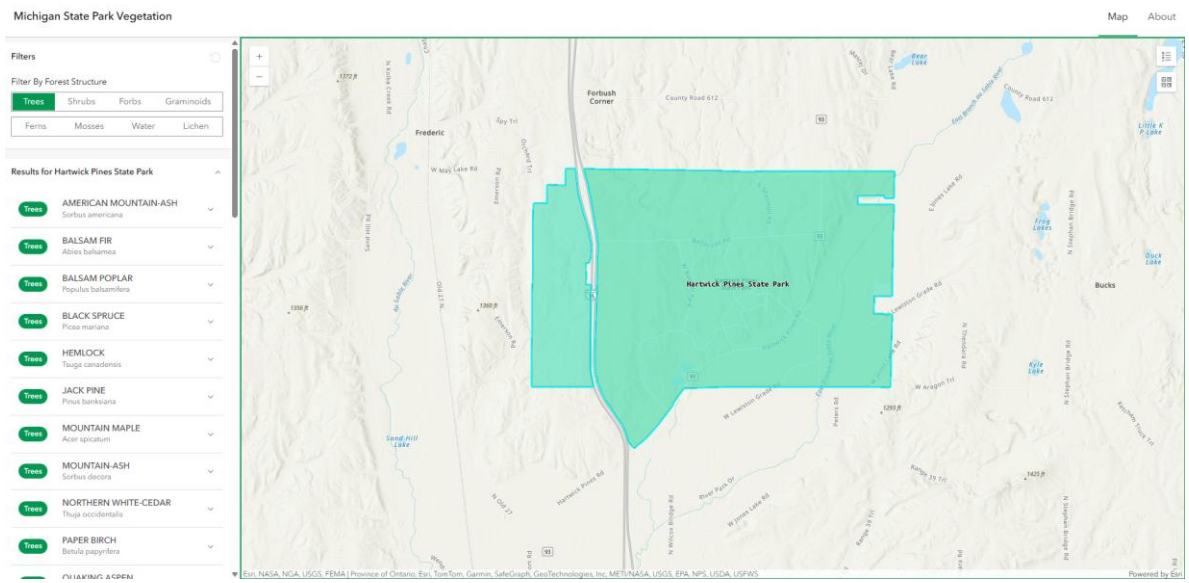

Figure 2. The User Interface

The user is given a slippy topographic map showing the state park boundaries and a side panel with filter functions and an area to be populated with results. The menu bar at the top links to lists of parks, additional information and links to related websites. The map gives the familiar options of panning, zooming, searching, and selecting while the additional controls are placed within the side panel. Once a park has been selected the species estimated to be found with the park appear as expandable blocks in a list showing their name, scientific name, and the forest structure location i.e. canopy, subcanopy, ground. Expanding the block gives images and descriptions along with an additional button to search the map for parks which the specific species is found in. Activating this clears the lists and repopulates it with a selection of all parks which that species is estimated to be found in, listing basic descriptions in the same style of blocks as the species. Clicking on one of these parks is analogous to selecting the park on the map.

The application exists as an HTML and ArcGIS Maps SDK for JavaScript frontend accessing an ArcGIS Online feature service (Figure 2). This feature service consists of three datasets organized within an ArcGIS Pro geodatabase:

the State Park boundary layer, the natural communities' layer, and the table of species. The natural communities and species have a relational class to satisfy the natural communities having multiple species contained within it. As these species can be related to multiple communities there must be a many-to-many relationship allowing selection in both directions. This is retained within the ArcGIS Online service and can then be queried. This query occurs by taking the intersection of a park boundary polygon and the natural communities shapefiles. By running the JavaScript API Relationship Query a feature array from the species table is returned. On the client side this can be looped through with JavaScript. The user interface is created primarily with the ESRI Calcite Design System adding HTML components such as panels and lists. These components are created as the script loops through each selected species and creates the expandable block containing the image, description, and search button. On the client side using DOM the displayed species can be filtered with the action buttons on the top of the interface. A full diagram of this process can be seen in Figure 3 below.

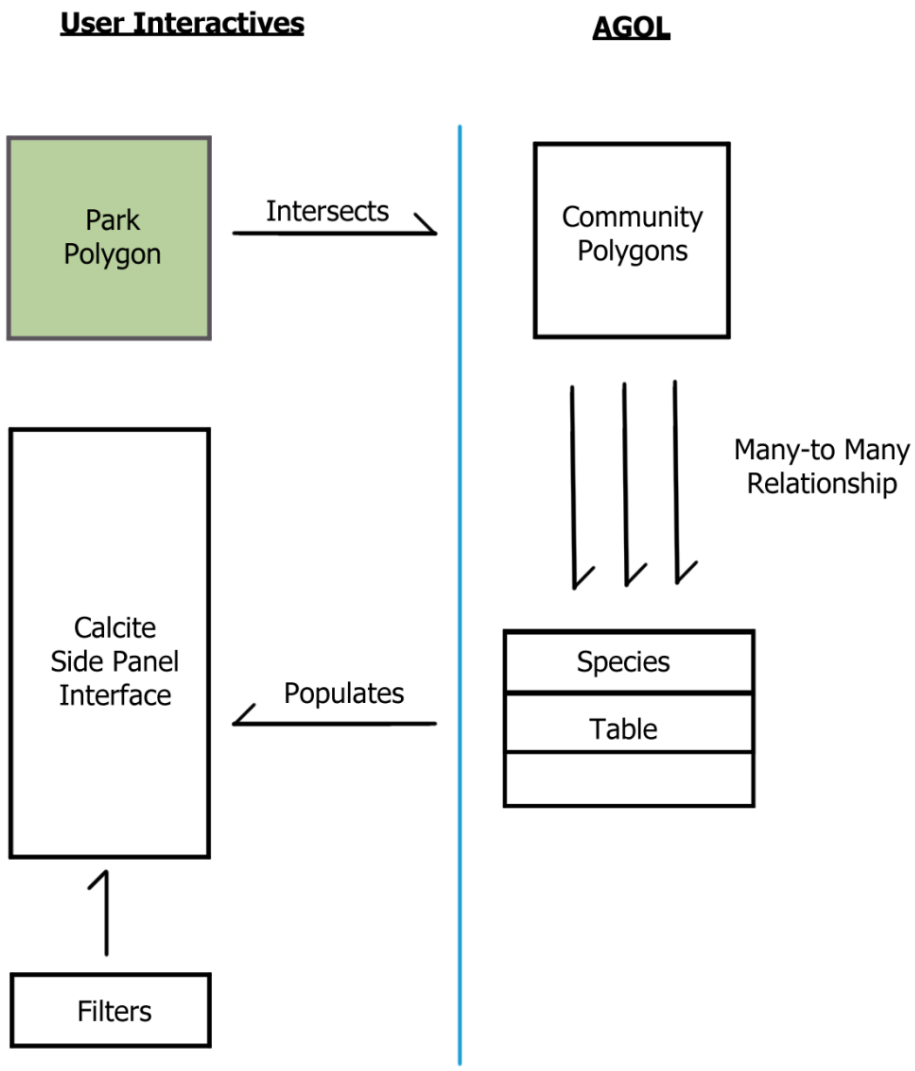

Figure 3. Diagram of the user interface and the actions occurring on the server.

This diagram shows the default method of query, but it is possible to message in the other direction, from the species in the interface querying all community polygons which intersect the park polygons. Every feature being independent allows for the backwards messaging as well as allowing for additions and modifications without affecting the other features.

Conclusion

The script is composed only of JavaScript and HTML/CSS however there had to be considerable consultation of the language and API documentation to create the intended UI/UX, while the database component was relatively simple to maintain. An issue encountered in the process was reliance on the JavaScript API for database query which when setting up the project appeared to satisfy all the conditions laid out and was able to perform them, however both the processing and amount of script required to carry this out was inefficient: the table feature had to be redefined as a layer without geometry in order for the function to accept it, and the array the query was returned in did not accept the built-in sorting and filtering methods of the API, these processes had to be created in the script itself e.g. removing all duplicates of a species which were returned. With explicit SQL functions many of these issues could have been avoided. Additionally there is heavy load times on larger queries which may be from both the method which the block elements are created or the server query method having high processing latency, with possible amends being aforementioned SQL query methods or changing the way the blocks load to occur immediately on the page loading instead of when the query is called, but these were unable to be implemented due to the time constraints and the extent to which they would alter the code.

Limitations on the project currently are the species information which is limited to major species due to scale and current reliable information sources. Adding this information would require considerably more time creating summaries from available literature and expert/academic consultation and communication. Similarly there is room for additional categories for each species such as going further into the characteristics like leaf or flow structure, associated species, etc. and additions of survey data, either from users or from land managers, to give a layer of accurate and location specific species.