Elyse Cheung-Sutton
GEOG 385
12/17/13

Finding Parties in Your Neighborhood in Davis

Summary:

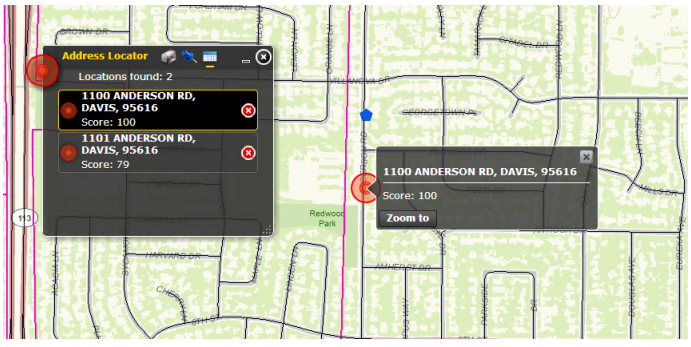
As Davis is a growing city with a growing college campus, new students flock to this city every year in hopes of a quality education and an amazing "college experience." For most students, parties are undoubtedly part of this college experience and will consume a good portion of their non-studious time over the next four years. Party hopping, a common term in the partying lingo, is the ability or the hope that a group of young socialites will be able to jump between multiple parties if one party is disappointing. There are many reasons a party can fail (the boy to girl ratio is unsatisfactory, the supply of booze has gone dry, or the atmosphere of the party is too ghetto/ratchet or too bro-y/douche-y) but good parties are not impossible to find if one is willing to put in the effort.

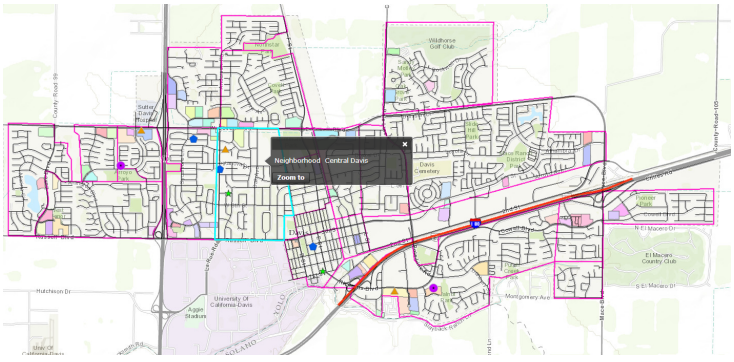
As social media has taken off in the recent years, useful phone apps can become invaluable to college students. The idea behind this app was to help groups of college students share information and take the stress out of finding parties.

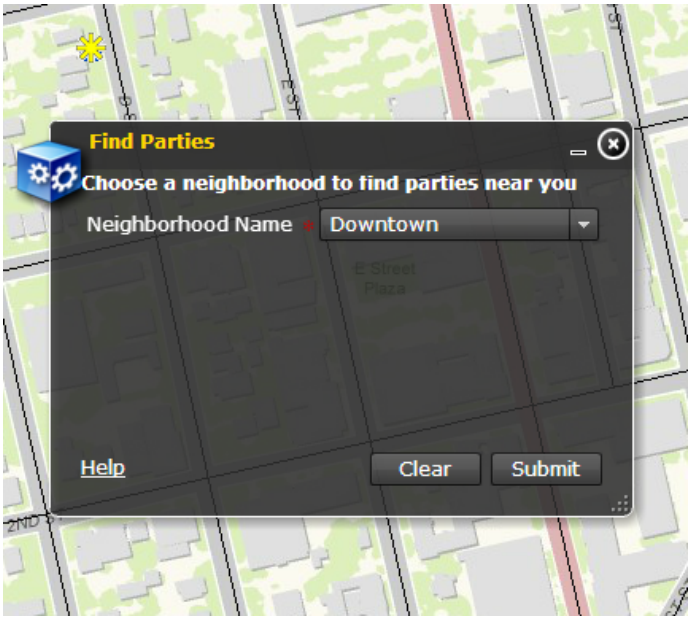
Purpose:

This map application was created as a space for sharing party information between a circle of friends. Similar to the way Google Docs works, a group of people could have access to a link to the web application. (A URL generator could possibly be used to create unique links in order to allow many groups to use the application. There would also be a need to create an empty feature class for the group to store their data.) Within that link, users could add party locations and other information in order to inform the group of upcoming parties. Users could then view these locations and search for near parties using their address as a starting point. This application would allow a group to share information conveniently and exclusively. By laying out the group's collective knowledge, decisions could be made quickly and easily. This system ideally would allow a big group of people to arrive at the same party thereby making the party more fun and successful. Also, if plans got changed (i.e. a party got cancelled), one could check the app to see what else was available and conveniently located.

The major task that I was not able to accomplish was to allow users to find parties based on their specific addresses. I thought that connecting the address locator to a geoprocessing tool was possible, but it turned out to be too complicated for the scope of this class. Thus I had to change the way users could search for parties by breaking this process into three steps. First, the user would input their address into the address locator. Next, using this location, the user would click on the map area to identify which neighborhood they lived in. Finally, the user could use that neighborhood name to search for parties near them.

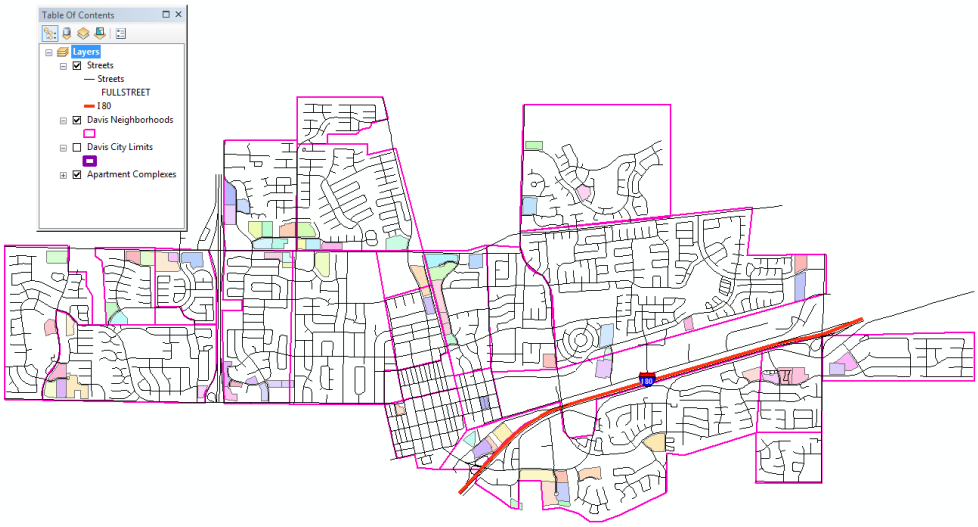




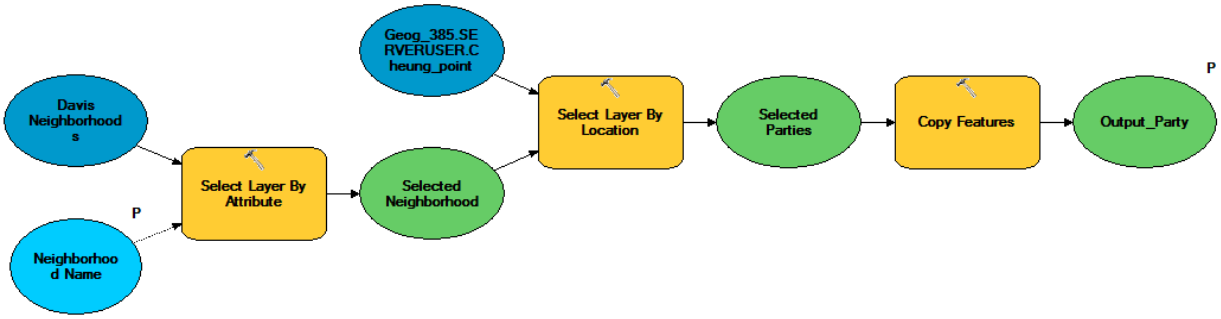


Description:
Data Used: Davis City Boundary, Davis "Traditional" Neighborhoods, Davis Street Centerline, and Davis Apartment Complexes
Software Used: ArcMap, ArcCatalog, ArcGIS for Server, Flex Builder

The first task to accomplish was to create a map service with a map that displayed Davis in a recognizable way. I found the Davis city boundary, an incomplete neighborhood layer, and an apartment complex layer on the city website. The neighborhood layer consisted of 4 small areas in Downtown Davis which comprised the "traditional" neighborhoods. Thus I had to fill in gaps myself by creating and naming polygons based on maps I found on Google and my own knowledge. In contrast, the apartment complex layer was extremely thorough and contained about 300 records. Since my goal was for college students to use the application, only apartments with over 100 units were displayed. This selection left about 100 complexes in my map. In reality, only about 10 apartment complexes are major party locations but I thought it was important to include as much apartment information as feasible since most students live in apartments. Next, I symbolized the street centerline file with the major streets as thin black lines and Interstate 80 as a thick red line and a highway shield symbol.

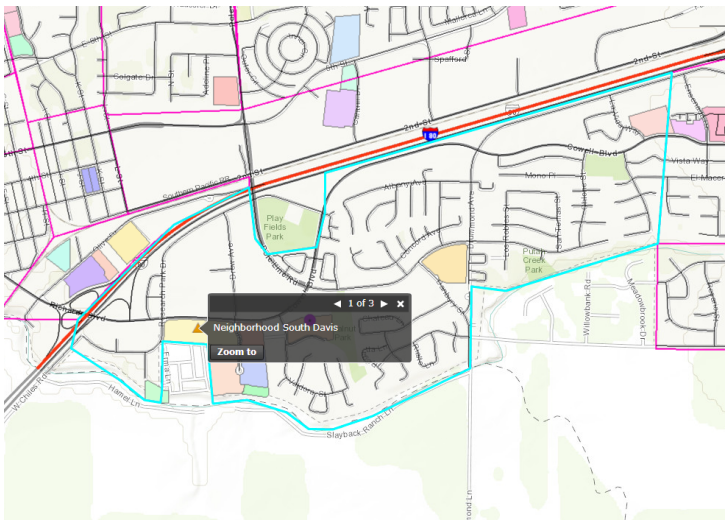


Next I had to create the services that would go with my web map. These were the geocoding service, the SDE hosted feature service, and the geoprocessing tool. The geocoding service was simple to create; I used the Davis street centerline which had the left and right addresses for the address locator. After publishing the address locator for the geocoding service to ArcGIS for Server, I linked the REST URL to the address locator widget in Flex and the service worked perfectly. I had the most problems creating the SDE hosted feature service for the feature editing capability in my application. The first step of editing my feature class to have the proper symbology and labels was simple. Creating a domain field and linking it to my feature class properly, however, took a lot of trial and error, including turning services off and playing with adding and editing fields inside ArcCatalog and ArcMap. (Honestly, thinking back on this, I still couldn't tell you which series of steps led to the working domain.) I then added and edited fields in the attribute table using ArcMap to ensure that the popup for the editing tool would look as desired. Once I had the domain and fields, I published the service to the Server and linked the REST URL to the Edit Widget. The final tool for my application involved creating a model so that users could search for parties (the SDE feature class) using their neighborhood location (an operational map layer).



Because we had done this in class, the creation of the model was very simple. My model consisted of the Select by Attribute, Select by Location, and Copy Features tool which allowed the user to input their neighborhood as the search parameter and that selection to be added to the map using the copy features tool.

Once the services were published to the server, Flex Builder was used to configure the web application. The map service included a topographic base map as well as the Davis map service and SDE feature classes as the operational layers. The three widgets used in the application were the address locator, the edit tool, and the geoprocessing tool. I also used Flex Builder to enable pop-ups (shown left) to allow the user to click around the map to identify what was being shown on the map. The pop-ups were not ideal because they consumed a lot of screen space and did not clearly identify what part of the map was being identified. The layout of the popup was confusing because, in order to identify a layer other than the one on top one, the user would have the click a small, obscure arrow at the top of the popup box. I would have liked something like the Identify Tool in ArcMap to allow a user to more clearly identify the map's layers.



Difficulties:

Creating the SDE hosted feature service was by far my biggest struggle. I was unclear about the components of an SDE feature class and how it worked; thus I needed a lot of instructor help to guide me through the properties. I was also fearful about messing up another student's domain which hindered my usual strategy of problem solving which involves clicking every box so I know which boxes give me which exact options. If I were to go back and repeat the process, it would take me just as long because I don't know which series of steps led to solving my problem.

Next, I really wanted the user to be able to locate parties using their address. However, connecting the address locator and the search function would require coding and extra knowledge that this class did not cover. My solution to this problem was to have users search for parties based on their neighborhoods. The user would first input their address into the address locator and click that area to identify the neighborhood. (Because Flex Builder did not have a simple identify tool, I enabled pop-ups for all of my layers. I did not like the method because clicking on the areas for identification might not be obvious to the user.) The user would then use the geoprocessing tool and input their neighborhood as the search parameter. The application would then show which parties were in that selected neighborhood. After seeing other projects, another way I could have done this was to create a search using a buffer tool. This would involve the user clicking somewhere on the map, a buffer to be drawn around that point, and parties to be selected based on that buffer.

Another small issue I had was with the legend. In my map application, I only used the point feature class from the SDE hosted feature services we created in class. The SDE feature class seemed to be bunched together in the legend and I couldn't figure out how to separate the point, line, and polygon layers so that only the point features would show in the legend. In the end, I did not solve this problem because I thought the legend was a minor part of my map application.

The last problem that I experienced many times was having to stop services in ArcGIS Server in order to edit parts of my project. I would try to edit part of my project thinking that I had all the services turned off and I would still get cryptic errors. I ended up just turning off all my services every time I wanted to edit something. This was annoying because I wanted to do lots of editing but would have to turn off all my services every time.

Final Thoughts:

I enjoyed creating this map application because the partying theme was very relevant to me. As a recent graduate, I ran into plenty of annoying problems looking for parties. I created this application with myself and other college students in mind which made it a more fun and involving process. Now that I have gone through the process of making a fun map application, I think that I am more ready to create an application that is relevant to science and "real life."

Although I really didn't encounter too many tough problems, there are many areas in which I felt that Flex was lacking which made my project less impressive. Like many other students mentioned during their presentation, there were many features I would have liked to add to my map application. The number one thing I wish that Flex allowed us to have was an instruction window. Even if a map is styled nicely and all the geoprocessing tools work, if the user can't easily navigate a site, a map application can be useless. My map application had an involved process consisting of multiple steps that had to be done in succession in order for the desired information to be displayed. If someone went to my application without any prior experience, most likely they would not understand how the application functioned. The other simple feature Flex lacked was an Identify tool. After seeing another student's project, perhaps the Search Tool would have been a good substitute.

If I were to continue to work on my application and add more features, I would begin by thinking about how to group data so that a user could share their information with only a certain group of

people. This would allow for information to be shared within a group of friends rather than with the entire online world. I would have to do a lot more research and probably learn some coding in order to create this accessibility restriction. As mentioned previously, I would also want a user to search for parties using an address and, taking it a step further, get driving directions to that location. I also should have included a map of the UC Davis campus as a good number of parties are held in college dorm rooms. I did make a few attempts at finding this data, but searching "UC Davis GIS data" returned too many results that were unrelated to the data I desired.

For a real web application, I would also have to think about data management. I would need a way for old data (i.e. a party that has passed) to be deleted. For the live data, I would also need a way to restrict the deletion of data. For example, a data point could only be deleted by the group's administrator or the user that created a point.

All in all, I thought that this project was difficult but not so difficult that I wanted to quit. I luckily did not run into many roadblocks and, after playing around, I was usually able to get my application to work. I hope that in the future, if I am required to create a web application, I will be able to remember all the very involved things we learned in this class.