Checklist:

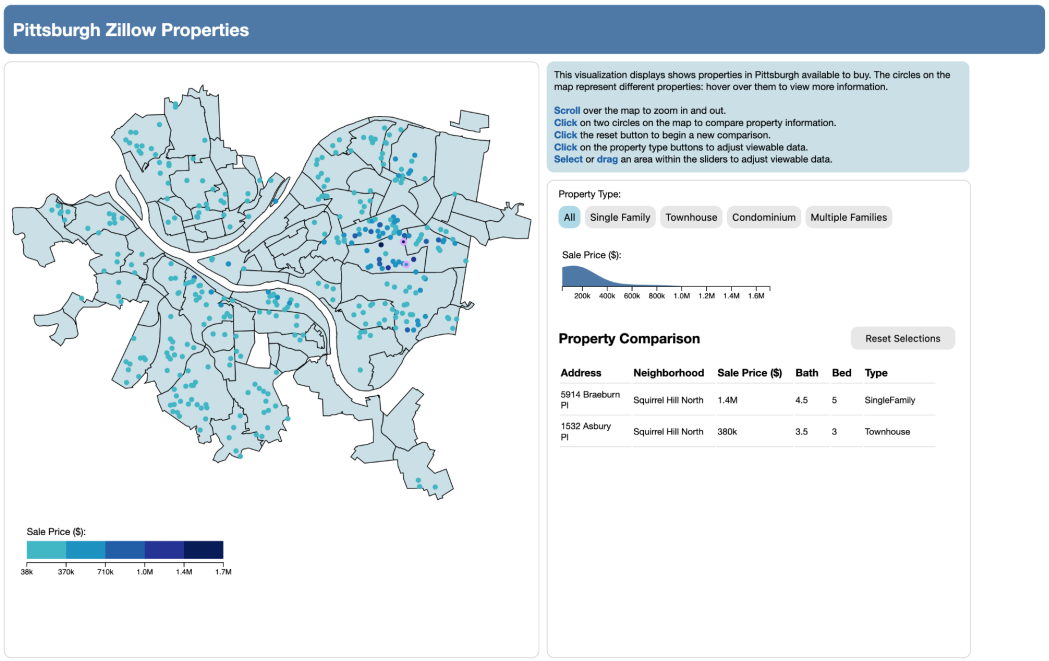
- [ ] Describe the data you chose and identify specific exploration needs that your user has
- [ ] Provide storyboards that outline the interactions you will design for your dataset and justify why you are using those particular interactions
- [ ] Briefly describe your final interactive visualization application, including a screenshot
- [ ] Step back and think about issues or trade-offs associated with the interactions you developed, and how you might alleviate those (or whether they are unavoidable).
- [ ] Briefly outline the development process of your tool. Explain how your visualization/interactions changed between storyboarding and final implementation. Comment on any trade-offs or design choices you had to make while developing.
- [ ] Identify how work was broken down in the group and explain each group member's contributions to the project. Give a rough breakdown of how much time you spent developing and which parts of the project took the most time.

*random thought - should we make there be a message when a user has only clicked one point and needs to click a second one?

Stuff not to forget:

- [ ] Paste in updated screenshot of viz when done
- [ ] Any other sketches to put into the storyboarding section
- [ ] Talk about how this fulfills the overview/details on demand mantra thing
- [ ] Talk about how it adapts flexibly and reversibly to user needs
- [ ] Time spent and contributions
- [ ] Finish any citations

Pittsburgh Zillow Properties



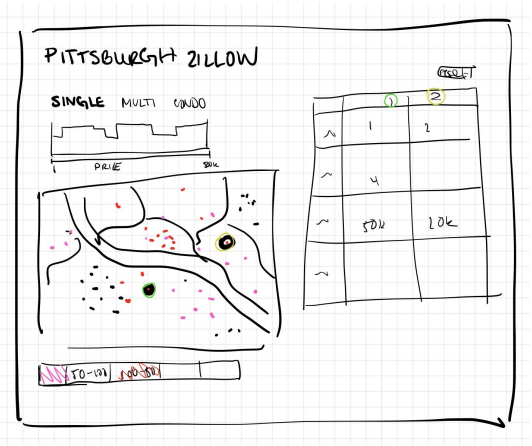
Data:

For this project, we decided to use the Zillow dataset that has data on Pittsburgh house listings. We used the filtered version of the data, as originally provided, and cut out any points that were missing data: we did, however, leave in some points that were missing bedroom or bathroom data, as we felt that our visualization would still be appropriate for users to explore even without it. We additionally took out one point that did not fit inside the Pittsburgh area that is visible on the topojson map, because it was technically not inside the area that this map considers the inner city (and was instead in the suburbs). We used most of the provided data, specifically the longitude and latitude for plotting the points, and the additional data for giving users detailed information in our panel/table view.

In this case, our user is someone who is looking to purchase a house in Pittsburgh, and is relatively familiar with the neighborhoods and overall map. As a future or current homeowner, a few of the most important information points to look at are location, price point of the property, bedroom and bathroom count, and property type, which we decided to focus on in our visualization.

Storyboarding Interactions:

Original sketches for our designs:



We considered many different options for how a user might navigate a map of houses, and finally settled on the base requirement that they would need to easily be able to compare two different properties to make this exploration meaningful for a buyer. For our minimum viable product, we decided on a few necessary actions. First, to implement a topojson map that had color coded points for housing depending on the price of the house. Second, a panel view with a table that compares the details of two houses. Lastly, filtering on property type and sale price of the various properties as noted in the provided data.

Final Interactions:
In our final visualization, we have several main interactions that help us to handle the high density of information provided by our dataset. The first is the map, which helps users to zoom in and out to aid point selections in order to see further detail, and the second is filtering, which aids the user to digest this data in the form of dynamic querying.

The first interaction is between the map and the table to the side. When a user clicks on two separate points on the map, the details of those two points will show up as a form of “details on demand” in the table to the right. When a circle is selected, the border color will change to signify this selection to the user. This will let the user compare the two houses in terms of street address, neighborhood, sale price, and number of bedrooms and bathrooms. The user can click the “Reset Table” button to clear the table of property information as well as unselect the circles on the map. The second part of the map’s interaction is a pan and zoom functionality, which first lets the user have an overview of the data (where they can view the entire map of Pittsburgh), and they can then zoom in and move around on neighborhoods they are specifically looking for, as they are likely familiar with the city.

The next section of interactions are the filters. There are two filters: one filter is for “Property Type” and the other is for “Sale Price”. To filter by property type, the user is able to click on the tab buttons to view all the properties or properties aligning in a specific category (e.g. Townhouses). Circles will appear or disappear based on the property type selected. The other filter for sale price and a histogram is displayed to allow the user to see the distribution of the price of properties. The user can then use the brush to select a range of the graph. The user can then slide this range around, expand or condense the slider range, or click outside the range to reset the filter, making sure their actions are flexible and reversible.

Development & Visualization Evolution:
In our initial brainstorming, we considered letting the user compare a property to the Pittsburgh average of data: this includes, sale price, number of bedrooms and bathrooms. However, there are many considerations in averages such as finding the average price per neighborhood, per property type, and more. Users may have different values in comparison to an “average” and so we iterated on our design to allow the user to compare two different properties, rather than an entire average.

Another development point was in our selection display. Initially we only displayed information in the table once a user had clicked on two points, but quickly realized that this was not comprehensive enough a way to give details on demand with a rapid response to an action. We then added in the capability so that when you mouse over a point on the map, the details show up in the table to the side so that a user can explore before making a permanent selection.

Other Trade-offs & Design Choices:

There were many data points, such as rental data, total rooms, tax information, finished size, and more. As there was a lot of data, we faced a trade off. We chose to narrow the scope of our audience to focus on buyers of property, not renters. Key information was selected to give users a high level overview of the various properties. While it is additionally a tradeoff to remove certain data points, as previously explained, they mainly contained unusable rows (i.e. more than one column was missing data), or were out of the bounds of the map (as in the one point outside of Pittsburgh bounds).

Evidently another tradeoff was the properties we let users filter by. Initially, we had discussed having more filters, including more statistics regarding pricing and other fields. While this type of advanced data may be useful to a buyer who is in the later stages of negotiations regarding their property, our visualization’s target audience consists of users who are still in the exploration phase of their homeownership. Rather than focusing too heavily on the statistical side of the houses, we instead wanted to focus more on the physicality of the houses/apartments (as in, bedrooms, bathrooms, property type, etc.) with quick information about the prices and locations.

It is also important to note that as we have learned in recent weeks, more filtering/information is not necessarily beneficial to the user: it can be much more productive to be purposeful with the data and actions we provide users, rather than trying to bombard them with options.

Another main tradeoff was the choice to only let users compare two properties. While it may be the case that certain buyers will need to compare multiple properties at once, visually two selections felt the most digestible for our audience. Instead of needing to glance back and forth between 2 or more rows of information about properties, the user's gaze does not need to go any further than quick glances up and down.

An additional tradeoff may be our implementation of the selection and filtering capabilities. We attempted to make the layout as simple as possible (i.e. a user only has limited point selection capabilities, and limited layered filter capabilities) to facilitate exploration. They can only select two points, and can use maximum two filters at once (sale price and property type). Despite this making the design visually simpler and easier to navigate, it may complicate the process of repeated filtering or selections. If a user wants to start a new "query," they need to reset the selections and then go back to the filters, rather than the table automatically clearing itself of selection. We did, however, think that the prospect of information being sticky and obvious to the user outweighed the need to change this design fully to accommodate the aforementioned issue.

Group Contributions:

Natasha Aysseh (nca28): Cleaned and processed the data; found and implemented the geojson/topojson file, into the final map with plotted points. Additionally worked on the map legend, styling, some debugging, and the writeup..

Amy Ouyang (aco49):

Gabriella Smith (gls245):

Time Spent:
In total, we spent around 1 hour on an initial storyboarding/design session, 4 hours on coding, 3.5 hours on styling, and 2 hours on debugging. We spent around 2 hours in total on the writeup.

Citations: (additionally marked in the code)

- makeSlider, pointPassesFilters function, credit to Prof. Jeff Rzeszotarski (code from INFO 3300 course, Lecture on Oct. 31 https://github.com/jeffrz/info3300-fa2022/blob/main/homefinder/static/index.html)

- drawLegend by Prof. Jeff Rzeszotarski (code from INFO 3300, Lecture on Oct. 3 https://github.com/jeffrz/info3300-fa2022/blob/main/notes/22.10.03.notes.htm)
- Geojson of Pittsburgh https://data.wprdc.org/dataset/neighborhoods2 (from Pennsylvania open data site) converted to topojson via https://mapshaper.org/.