Describe the data you chose and identify specific exploration needs that your user has

We have selected the Zillow dataset, which provides comprehensive details on the properties available for sale in Pittsburgh. Our intended audience comprises individuals who are looking to find their ideal homes in Pittsburgh, Pennsylvania, such as those who are planning to move to Pittsburgh or those who are searching for a new residence within the city. To identify the most suitable property, users may need to explore various factors, such as the type of property (single or multiple), the number of bedrooms and bathrooms, and the price range for sale or rent.

In addition to the Zillow dataset, we also found a GEOjson dataset of neighborhoods within Pittsburgh. This dataset was essentially to create smaller divisions within the whole of Pittsburgh in which users can select specific neighborhoods that they want to live in or just interest them in general. It's through these divisions that another sort of "filter" is created aside from the ones we coded, where users can navigate through different neighborhoods without having a specific filter to fill out. Another thing to note is that one of the data points from the original Zillow dataset had to be removed because it wasn't located directly in Pittsburgh and therefore couldn't be placed on the map. When implementing the dataset the point was placed a bit far off the map, so we decided to remove the point as a whole, as our goal was to help users select houses that were based in Pittsburgh and only Pittsburgh, so it wouldn't make sense to have any other random points.

Provide storyboards that outline the interactions you will design for your dataset and justify why you are using those particular interactions

[Filter]

To help users find houses that meet their specific needs and preferences, the filtering system allows for the selection of one or more property types, as this feature is not strictly defined like bedroom numbers or price ranges. Users can also select the number of bedrooms, bathrooms, and total rooms using the filter, with bedroom and bathroom selection being preferred to filter more data and enable users to focus on the features that matter most to them. Total rooms can be entered using a text input since presenting all options in radio buttons or a dropdown menu could overwhelm or distract users. For the price range, users can indicate whether it is for sale or rent, and the corresponding bar chart is displayed on the form. Once minimum and maximum prices are entered, the system filters and displays houses within that price range on the map. Finally, users can submit their filter by clicking the "apply filter" button, with a reset button added to enable quicker creation of new filters since radio buttons cannot be unselected. The submit button is deemed more efficient and practical since users may apply multiple filters.

[Price Range Bar]

Users can get a sense of the price range from the bar chart provided. Users can refer to numbers in the x-axis when they enter the minimum and maximum prices. By looking at the y-axis, the height of the bar chart, users can be aware of the number of houses in a specific price range. The default setting is “For Sale” and once users click on “For Rent,” the axis and bar of

the bar chart change to the price range of rents. We chose the radio button because there are only 2 options and users must select 1 item to filter.

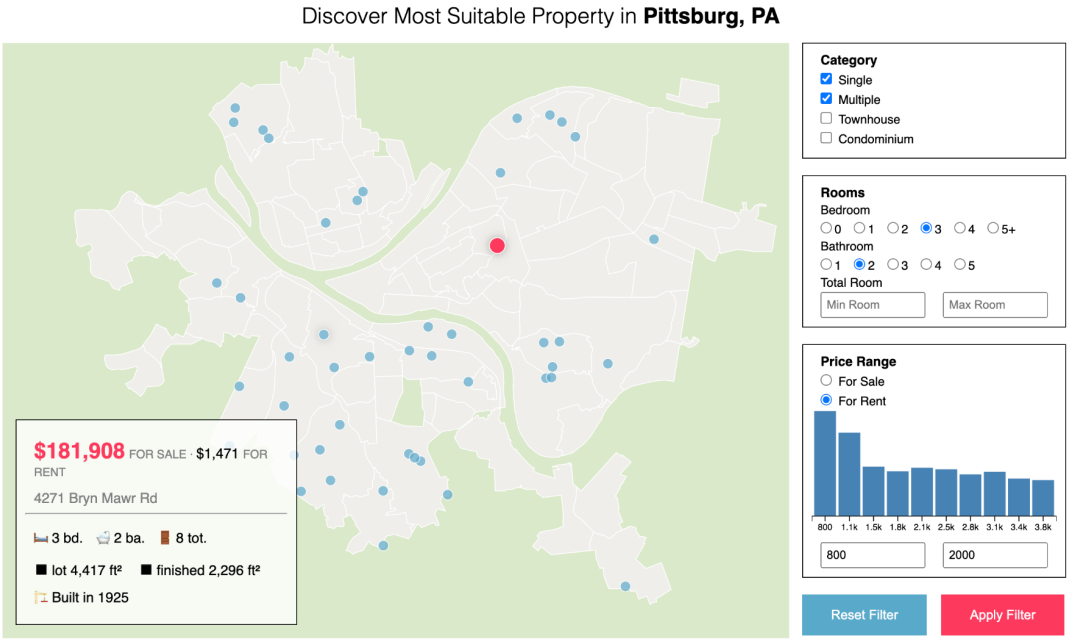
[Information box]

When users click a data point on the map, detailed information about the house appears on the right side of the map. The detailed information includes: street address, sales price, rent price, number of bathrooms, bedrooms, total rooms, lot and finished size, and the year built in. We chose to click as an interaction to let users quickly obtain relevant data along the points as well as keep track of a house they clicked on if they were to check out other parts of the map.

[Pan and Zoom]

Since there were a handful of data points on a somewhat small map, we wanted to make sure users would be able to differentiate between houses that were very close to each other. We decided to implement two different types of zoom. The general panning was necessary for such a map for users to be able to look closely at the separate houses in close proximity and scroll over to different neighborhoods. The two types of zooming, on the other hand, were used to give users more of a sense of choice and customization. The first type of zoom is the basic one, where users can scroll in or out in order to get closer to the specific area they're interested in. The other type of zoom allows users to click on the different neighborhoods that exist in Pittsburgh and zoom directly into that neighborhood. If a user desires to live in a specific area they're aware of and know where it is, they can click on said neighborhood, displaying them the name of the neighborhood and immediately zooming in on the cluster of houses available in said neighborhood.

Briefly describe your final interactive visualization application, including a screenshot



[Filter]

The filter system can help users find houses that meet their specific needs and preferences. It allows users to select one or more property types, the number of bedrooms, bathrooms, and total rooms, and a price range for either sale or rent. Once users submit their filters, the houses on the map that satisfy the filter will remain while the others will disappear. Users can adjust their filter or reset the filter based on their needs.

[Price Range Bar]

When users select either For Sale or For Rent, the corresponding price range bar chart appears. It has the price formatted in 1 decimal point with k and m signifiers and the count of the house in that range on the y-axis.

[Information box & clicking on the map]

When users click on the data point on the map, the color of the circle changes to red. Then, the box that contains detailed house information appears in the left bottom corner of the map. When not selected, the information box does not contain specific information and prints a "Click a circle to view details!" message.

[Pan and Zoom]

Users will be able to click on the various neighborhoods on the map, allowing animation to play out and an automatic zoom onto the neighborhood, displaying mostly the available houses in that neighborhood. They will also be able to generally scroll in and out to do a basic zoom in and out. The panning is also basic and allows users to click and drag the map in order to move around it.

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).

[Information box & clicking on the map]

Because the information box is oriented on top of the map, it covers some data points. The trade-off is that users can quickly read the information on one screen because it is close to the data point selected, but they have to move around the map to select the data point that is covered by the box. We could have placed filters above the map and placed the information box on the right side of the map so that the information box is not overlapped. However, the filter area was a bit overwhelming to be on top of the map and had large whitespace on the right side when we placed the information box solely on the right.

[Display ranking of the house]

We initially brainstormed an idea to display the ranking of the houses based on the filter. For instance, given the price range, display the top five houses within the price range, have the largest finished size and lot size, and the latest year built. We realized that because we are displaying the houses on the map, showing the ranking would not be effective. Also, with the appropriate amount of filters, the number of houses on the map is not large. We decided that the users would be able to efficiently explore the dataset even without the ranking.

Briefly outline of 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.

[Filter]

We created a form that takes multiple inputs. The type of input is chosen based on the value (more detail are explained above). We allowed empty inputs to give users more flexibility. Thus, a value of a variable is null without input. We attach event listeners to the buttons to allow submission or reset of the form. We also varied the display of the bar chart based on the checked radio button. The filtering is done by filtering the actual Zillow dataset. Then, we created and used the function to re-render the map afterward.

[Price Range Bar]

For the price range bar, the width of the filter area was narrow so it was unable to put all of the axis descriptions. Thus, we formatted the y-axis in the ".2s" format to give an organized look. We chose 1 decimal place because we could not see the difference between the axis when we round up to the integer.

[Information box]

For the house information box, we decided to put it on top of the map instead of the right side of the map. The filtering area was long to fit in the information box below it in one screen. Thus, we moved it to the map so that users can directly see the results quickly. We reduced the opacity of the information box to make the data points behind it still visible. We visually prioritized the price information inside the box with the red accent color as we thought the price is the most important criteria for people who look forward to the houses. We used emojis to organize the information visually. As we decided to put the information box on top of the map, we thought it would be inconvenient to display information when hovering because users have to drag over the map to select different data points. Thus, we changed the interaction from hovering to clicking so that users can still have the information they have selected while they move over the map.

[Neighborhood Name Display]

At first we thought it would be a good idea to create an additional filter for the different neighborhoods, allowing for the user to choose a specific neighborhood and after applying the filter, the map would zoom in on that neighborhood. After working around in the project for a bit, we decided against this idea and instead allowed for the neighborhood zooming-in to happen on the user's click. Our initial idea was to have something like a dropdown to be available for all the neighborhoods, but the idea of putting a dropdown with approximately one hundred options seemed very inefficient and would clutter the general filter space we had available. In addition to this a lot of the neighborhoods in the map didn't contain any available houses from the dataset, so having the function to filter through all the neighborhoods, with a handful not having any houses seems unnecessary. So instead users would have to click around in some of the main areas in order to find a specific neighborhood they're interested in, which seemed more efficient.

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.

Yubin:

- Implement the filter and rendering map based on the filter (~5hr)
    - Filter the dataset by price ranges
        - Enable selecting the type of price (ex. sales or rent)
    - Filter by bedroom and/or bathroom
    - Filter by total rooms
- Styling the filter and the map (~2hr)
- Sort to find the top 5 houses and filter by property type (~ 1hr)
- Contributed to final writeup (~1hr)

Jessie:

- She created and styled the information box about the house when the point on the map is clicked and connected with the map interaction. She also developed bar charts for the price range filter and formatted the map, information box, and the filter area to fit in. She also contributed and edited the final writeup

Chris:

- Created basic map (~1hr)
- Implemented both zooming types and the panning (~2hr)
- Implemented part of click interaction with data points (~1hr)
- Plotted data points on the map (~1/2hr)

- Contributed to final writeup (~1hr)