**PokémonStay**

**Musa Capstone - Ziyuan Cai**

1. **Motivation**

The map of Pokémon Go as shown in figure 1.1 contains nothing but only street map, which makes it impossible to know what exactly the street or road is. Pokémon players currently locate themselves by switching back and forth between Pokémon Go game and Google Map, when they are not very familiar with where they are. Also, because restriction of the user interface, it’s hard for players to know the situation 3 blocks away even in the game. In that case, they will hardly know where to go to catch Pokémon safer and faster. So, I want to build an interface for the players to locate themselves and find their way to the best place to catch Pokémon. Due to the limited time and technology, I am going to focus on Philadelphia first and build a wireframe for the newcomers as well as the players in Philly for a better game experience.

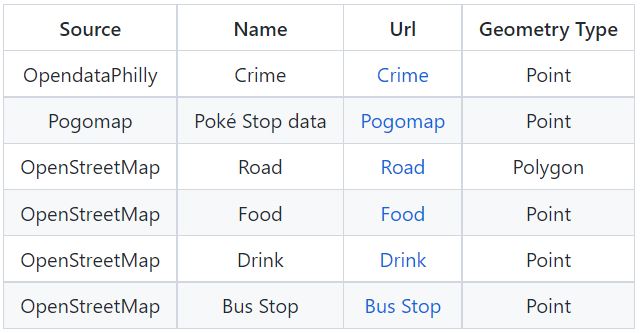
Figure 1.1 Pokémon Go



1. **Data**

It’s worth mentioning that based on game settings Pokémon will show around Poké Stops, which means if there is more Poké Stops in an area, there will be more Pokémon. In the game world, there are four kinds of Poké Stops, basic Poké Stops, Gyms, Ex Raids and Nested Pokémon. Except the basic one, all the three advanced Poké Stop have different functions in the game, so we need to take it into consideration.

Figure 2.1 Data

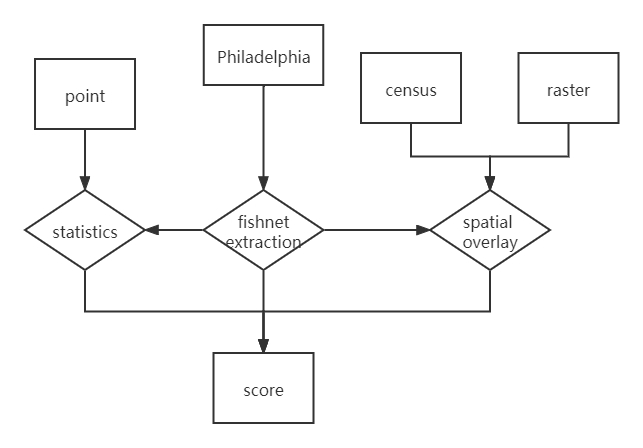
****

The Poké Stops data is from the Pogomap Website (<https://www.pogomap.info/>). First, I expect to use web scrapping for this data. After trying several methods and realizing the website is under protection, I got the data manually. All the other data is the description of the real world, containing all the crime in Philly from 2016 to 2021; tags related to food, *Restaurant, Fast Food and Food Court*; tags related to drink, *Pub, Bar and Café*; all the bus stops.

1. **Method**

I would like to divide this section into two parts, analysis part and wireframe part. For the analysis part, considering the difficulty of implementation, I did a tricky operation that set the destination to be solid points. In my expectation, destinations should be removable and random, which is more in line with our understanding of how a real-time map works. However, if we introduce the thought of limit in mathematics into it that the points set in advance are far more enough and the distance between each point is small enough to fill the whole map, it will reach the same result with a real-time map.

Figure 3.1 Flow



So, as shown in the flow chart that I first cut Philadelphia into 500 meters’ fishnet and then set each centroid of the grid to be the final destination. Based on the equation, I did calculation on each destination points, figuring out which point would be the final point shown in my website for the users.

This formula is formed by myself, where means destinations, means the distance between the start point and each destination. and are the continuous variables in the analysis, which means the total number of Poké Stops and the number of crimes from 2016-2021 within the 500 meters buffer zone of each destination. To fit the analysis, I normalized the two continuous factors into the range from 0 to 1. And all the factors left, , , , , and , are binary variables, showing whether there is any *food*, *drink*, *bus stops*, *Gyms*, *Ex Raids* or *Nested Pokémon* within the 500 meters buffer zone. 0 means there isn’t while 1 means there is. For the parameters, they were set rely on the experience of my own game life in Pokémon Go and the continuous attempts. I wish the users to find somewhere near them to catch Pokémon, so I give 5 to the factor distance. On the other hand, the number of Poké Stops have already weighted too much in the analysis and that’s the reason I gave it a parameter of 2.

In terms of the wireframe, I used html design and JavaScript language to build my website. Although the data of Poké Stops keeps updating, compared to the existed data, the new data is only a very small part. As a result, I set my website as a stable version and used geoJson file including my whole analysis and results as input to my website. For web implementation, points adding and route drawing, I used a map tool called Leaflet, which is really easy.

1. **Result**

This section will also be divided into analysis part and web tool part.

Figure 4.1 Fishnet Figure 4.2 Destination

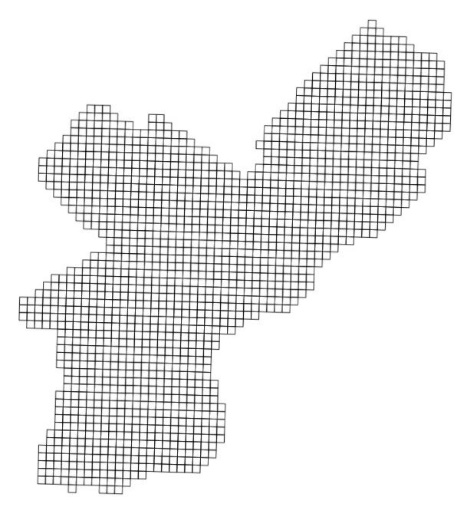
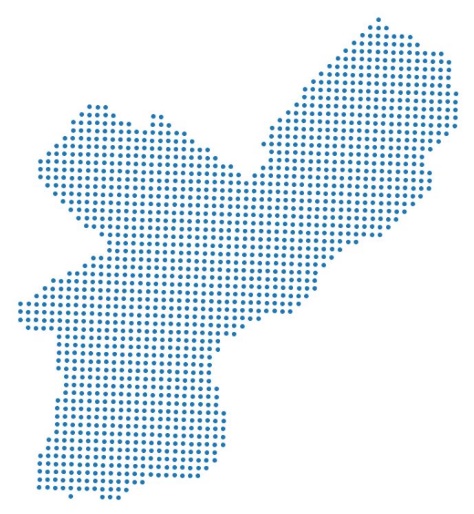
 

Figure 4.3 Crime Figure 4.4 Poké Stop

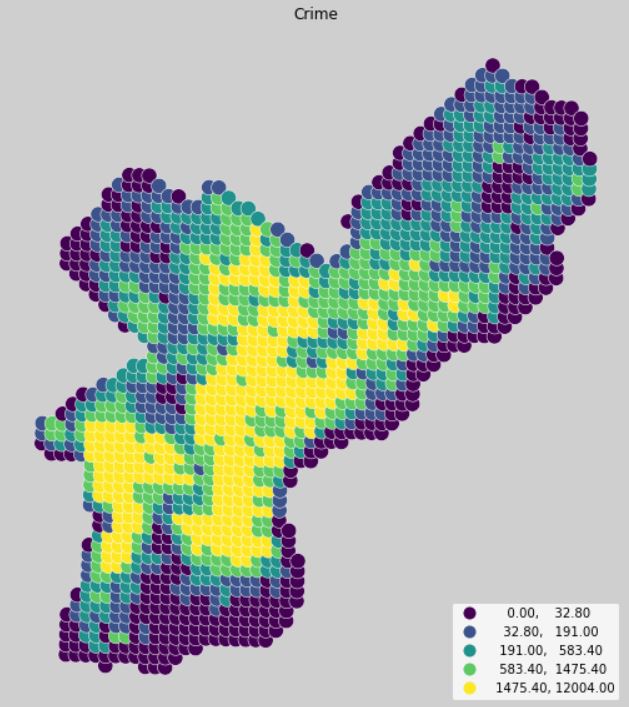
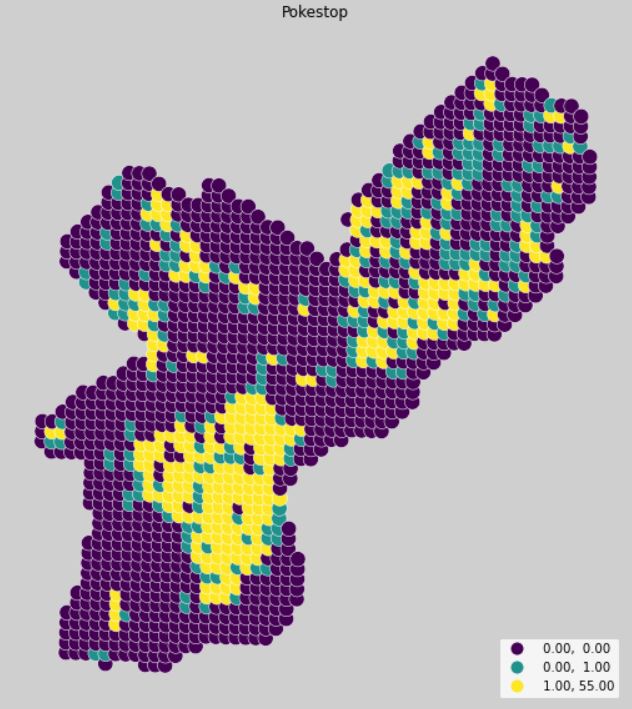
 

Figure 4.5 Food Figure 4.6 Drink

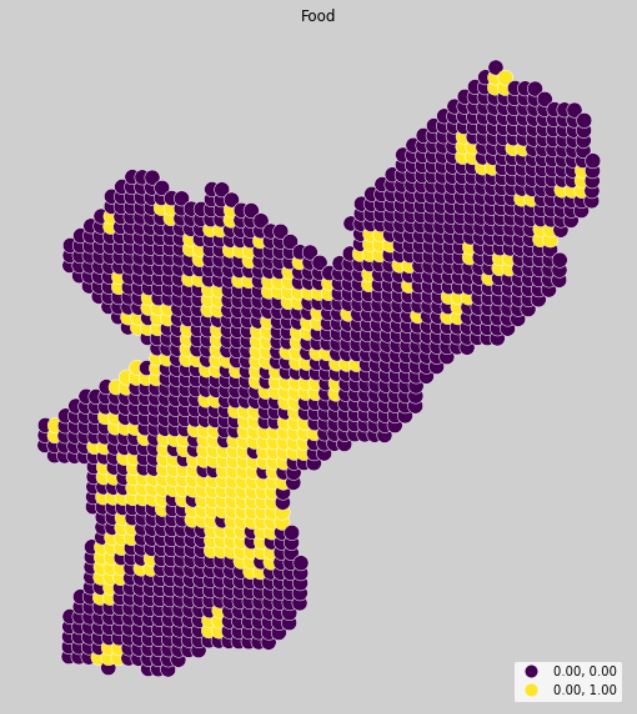
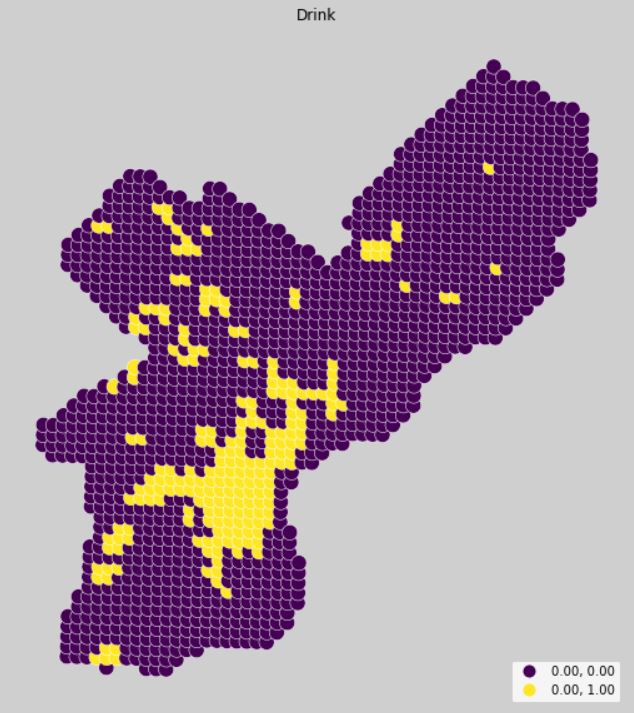
 

Figure 4.7 Bus Stop Figure 4.8 Gym

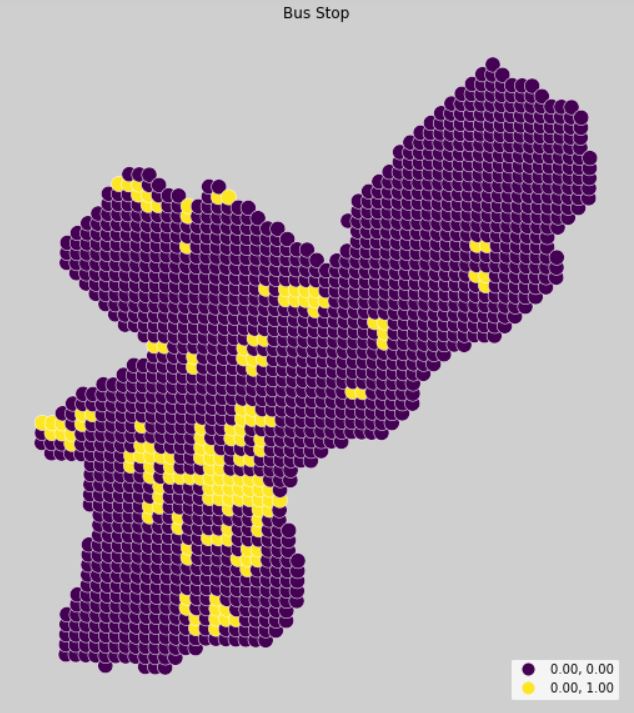
 

Figure 4.9 Nested Pokémon Figure 4.10 Ex Raid

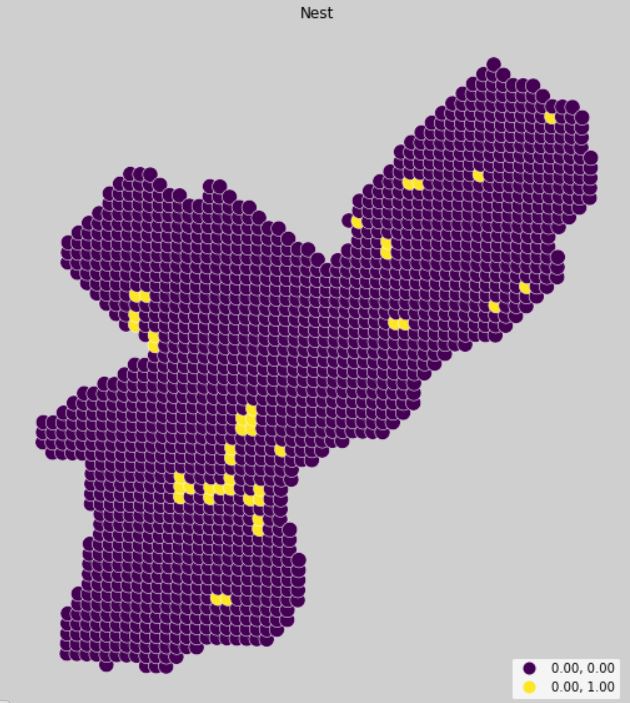
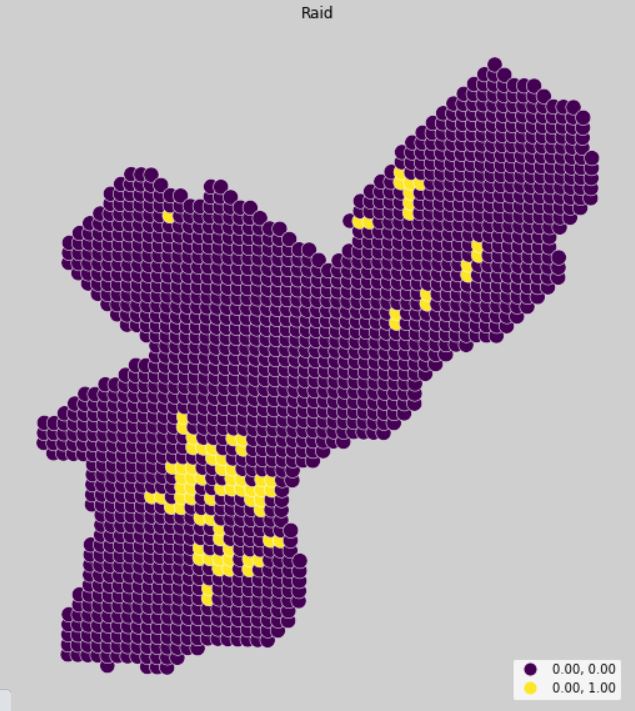
 

Figure 4.11 Start and End Figure 4.12 Route

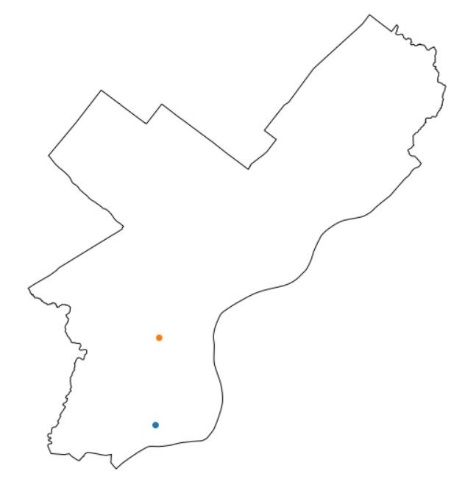
 

Figure 4.1 and 4.2 are the results I got really early, as they are the basis of the whole project. From figure 4.3 to 4.10, I show the results of all the factors, where each dot means the 500-meter buffer zone. In figure 4.11, the blue point was the random start point I chose for trail and adjustment of analysis while the orange point was the destination after calculation. The route is given in figure 4.12 by doing web analysis with the package of OMS.

Figure 4.13 PokémonStay

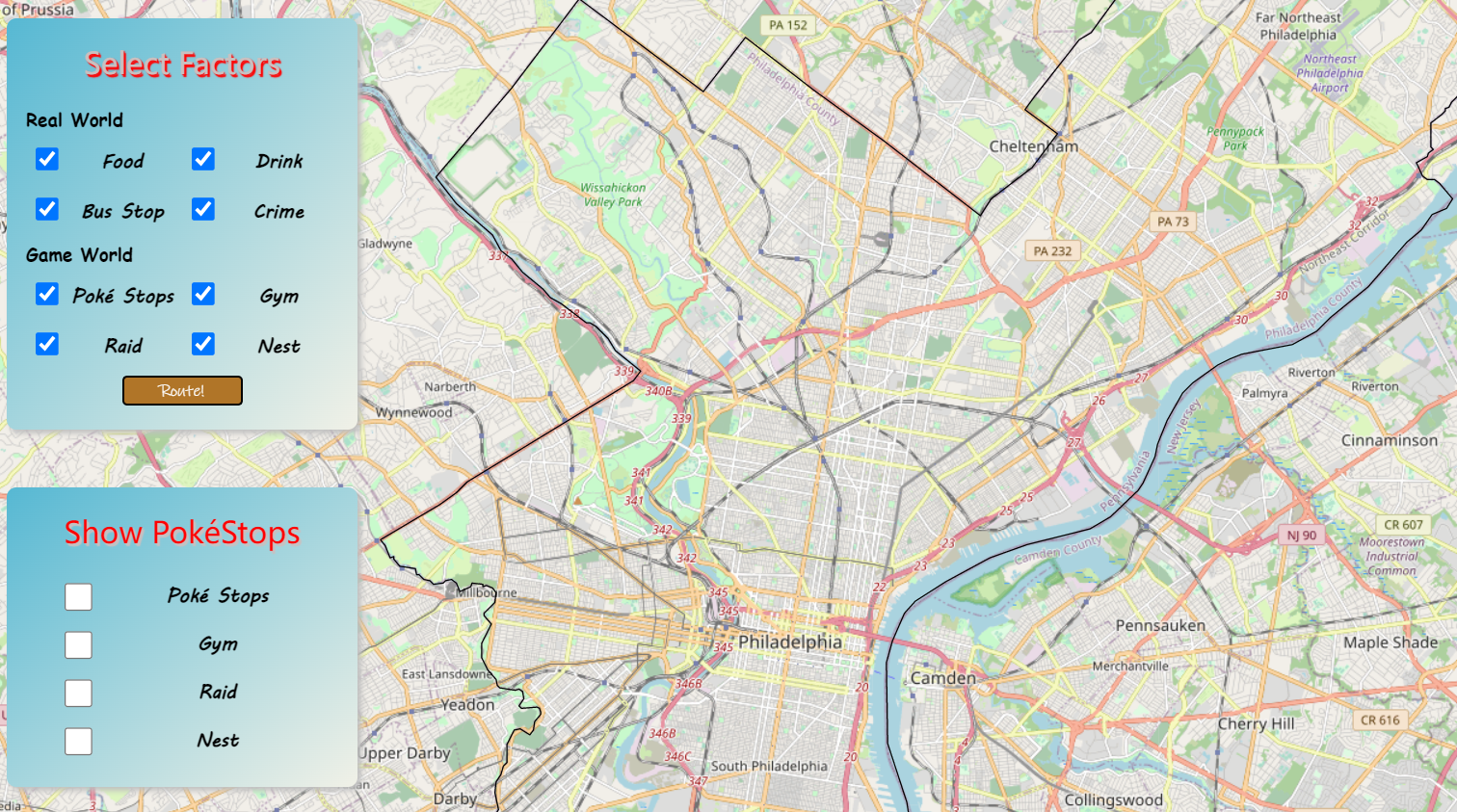


Figure 4.14 Show Case

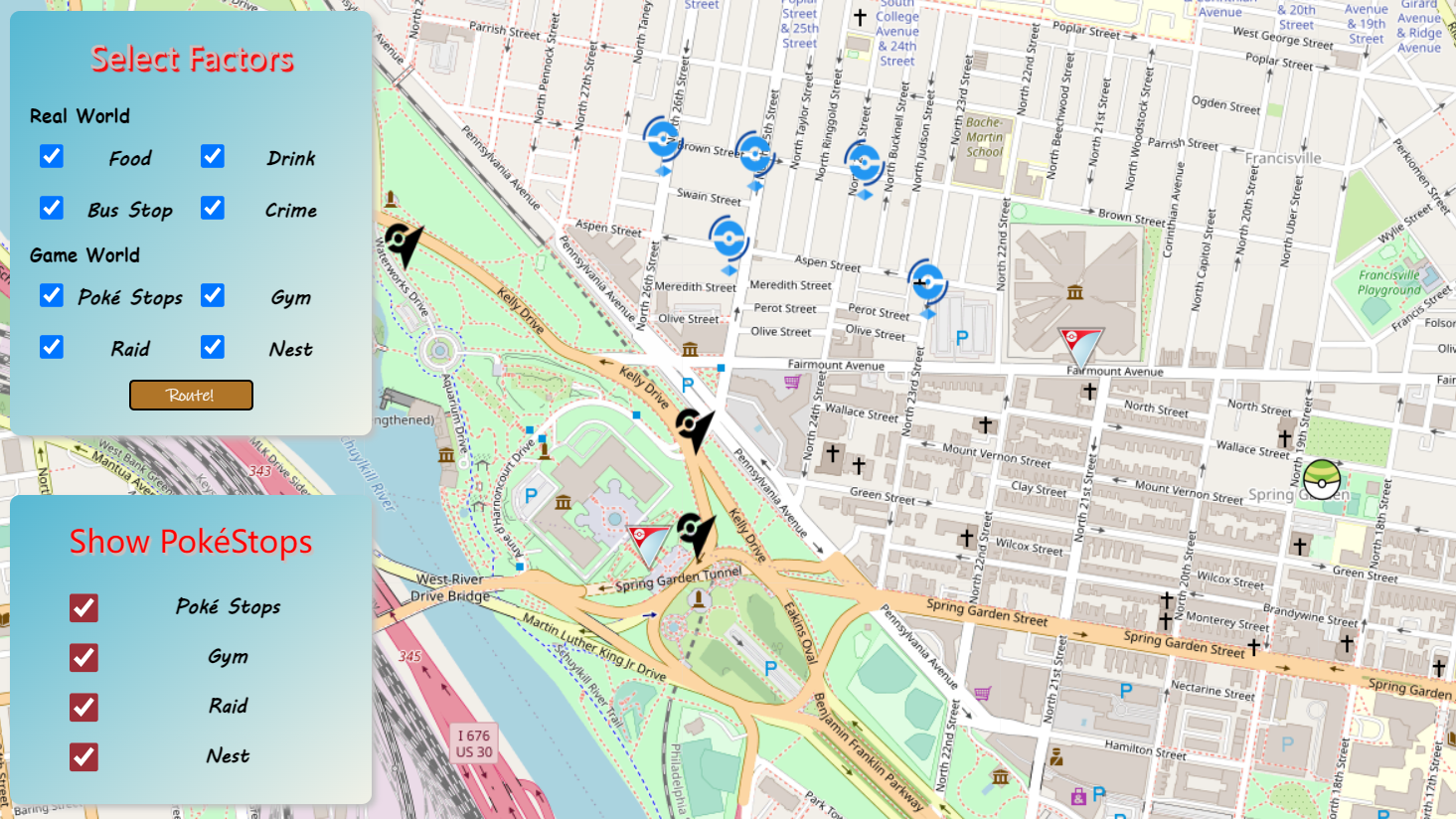
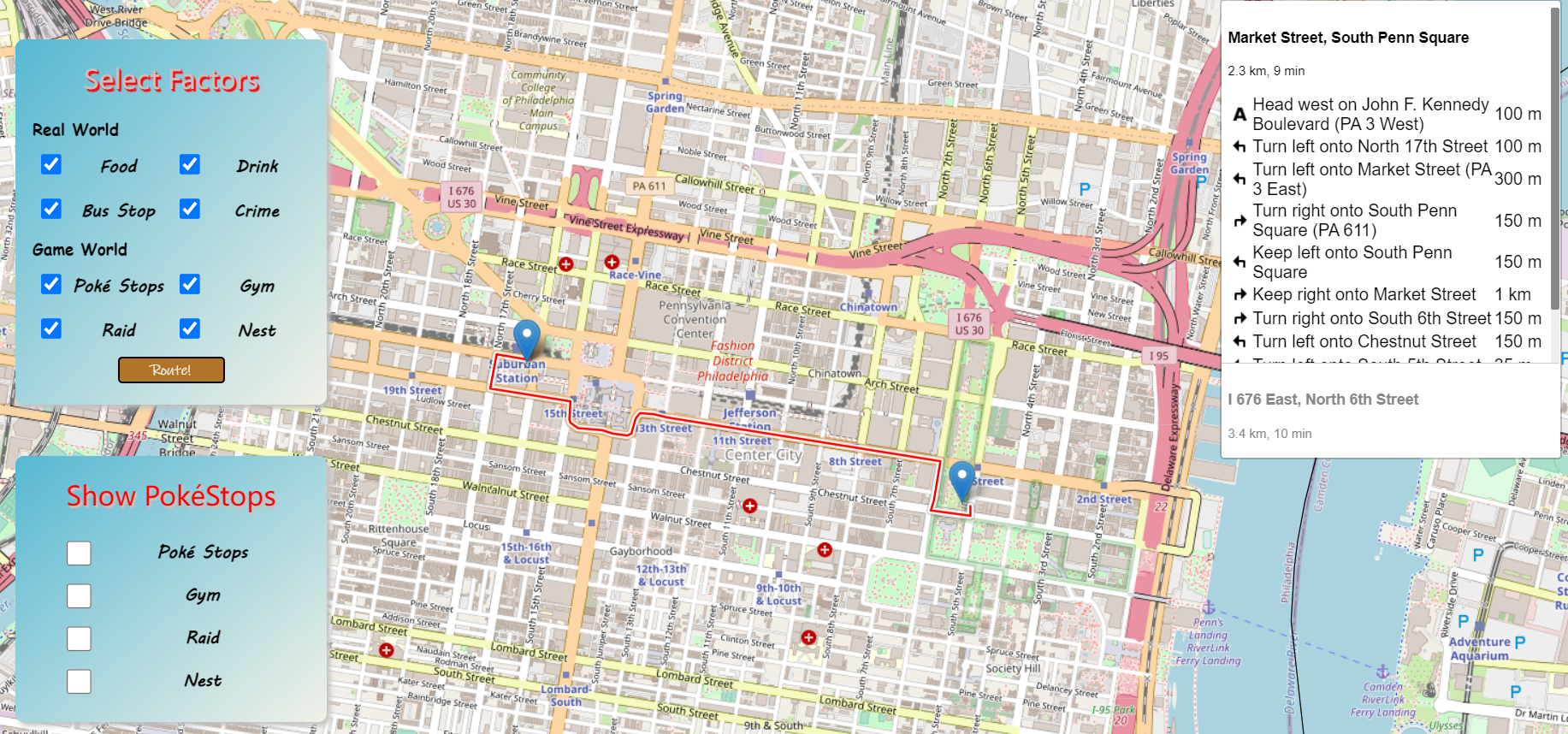


Figure 4.15 Route



This is my final PokémonStay website (<https://krmeteor.github.io/Pokemon-Stay/>). Figure 4.13 shows the sight users first go into the page. Figure 4.14 shows how to use the second board on the left which is for displaying all the Poké Stops, where blue balls are “Poké Stops”, red inverted triangles are “Gym”, black icons are “Raid” and green Pokéballs are “Nest”. Users can easily set the start point of where they are by clicking on the map, select the factors they want to take into consideration in the first board on the left and the click “Route!” for the route to the calculated destination. In the last figure, users can also follow the detailed navigation on the right.

1. **Conclusion**

Overall, I achieved what I wanted to do. For me, this is really an interesting and challenging project, also the first large-scale project completed by myself. However, my final result has some flaws. In my final testing phase, I found that no matter where the starting point is or what the factor selections are, the destination is always a few fixed points. I carefully compared the Pokémon Go game and my website, I found that my data is a lot less than the actual data, which is the main reason that my results are not so reasonable. If I got time to improve this project in the future, I would focus on the improvement of the data at my best.