

PokeMap: An Interactive Pokémon Map for Finding Encounter Data

John Hulton, Jared Okun
Computer Science Department
Rensselaer Polytechnic Institute
(hultoj, okunj) @rpi.edu

1. Introduction

The Pokémon video games are some of our most-loved video games. We have been playing them for over half of our lives. A key part of the main series of Pokémon games is running into and catching wild Pokémon. In most locations in the game, as you move around, you will have random encounters with random Pokémon. Every location in the game (where you can randomly encounter Pokémon) has a set of Pokémon that you could encounter. Each Pokémon in that set is assigned a percentage chance that you encounter it. We call this set and the corresponding percentages a Pokémon Encounter Distribution (PED). An example of a PED is at the location Route 201 in the game *Pokémon Platinum* most players will have a 50% chance of finding the Pokémon Starly and a 50% of finding the Pokémon Bidoof. A player can catch these Pokémon to add them to their team, so players care which Pokemon they can run into and will often seek out certain Pokémon to catch.

1.2 Motivation & Audience

PEDs initially sound simple, but there are two complicating factors: some locations are divided into subsections and each subsection has a different PED and a location's PED can change based on certain encounter parameters. For example on Route 201, if your system time is 4:00 a.m. – 9:59 a.m. (morning) or 6:00 p.m. – 3:59 a.m. (night), there is a 10% chance to encounter the Pokémon Kricketot; if you are using the in-game item called the PokeRadar, there is a 5% chance to encounter the Pokémon Nidoran M and a 5% chance to encounter the Pokémon Nidoran F; if you insert the GameBoy Advanced Game *Pokémon FireRed* into a Nintendo DS console while playing *Pokémon Platinum* on that console, there is a 8% chance to encounter the Pokémon Growlithe. For each of these parameters, the chances that you encounter other Pokémon are adjusted so that the total percentage is always 100%. There are several

PEDs for the location Route 201, containing between two to seven Pokémon. The problem is, the game doesn't tell you any of this; you don't know which Pokémon you can encounter in a location until you encounter them. Given the amount of different encounter parameters and the low chances that you run into some Pokémon, the game makes it very difficult to actually know what you could catch in each location.

The purpose of our visualization, *PokeMap*, is to give our users a simplistic tool to figure out which Pokémon they can encounter in each location in the game *Pokémon Platinum*. A user can view the different PEDs of a specific location by toggling the encounter parameters so that they can see what is needed to encounter certain Pokémon. A user can also mouse over a location if they want to know all of the Pokémon that could be encountered in that location. The tool should be useful to all players of *Pokémon Platinum*, but there are three different groups we focus on: casual or new players, who want to see what they could encounter as they progress through the game; Nuzlockers, who are players that do a special challenge run where the first Pokémon they encounter in each location is the most important; and speed runners, who would want to a way to find many PEDs very quickly.

1.2 Research Question & Hypotheses

Pokémon Platinum has been available since 2008, so there are several other tools that can be used to see what Pokémon are available on each route. Our research question is simply: Is *PokeMap* a more effective tool for finding PEDs than its contemporaries? Two of the most popular contemporaries are the websites *Serebii.net* and *Bulbapedia*. Our hypotheses are:

1. Users will be able to find the PEDs and Pokémon in each location easier with *PokeMap* than *Serebii.net* or *Bulbapedia*.
2. Users will be able to find the PEDs and the Pokémon in each location faster with *PokeMap* than *Serebii.net* or *Bulbapedia*.
3. Users will better remember the PEDs and the Pokémon in each location when they are found with *PokeMap* than when they are found with *Serebii.net* or *Bulbapedia*.

2. Background Material

2.1 Similar tools

As mentioned above, two tools similar to *PokeMap* are *Serebii.net* and *Bulbapedia*. On *Serebii.net*, a user can select a particular location from a drop-down menu and see each of partial PEDs for that location. *Serebii.net* does not show how the total percent chance is lowered to 100% after the encounter parameters

are applied. Serebii.net also mixes the data for *Pokémon Platinum* with the data for the games *Pokémon Diamond* and *Pokémon Pearl* (the map of each game is nearly identical) and includes additional data about each Pokémon. Figure 1 shows a screenshot of around one-fifth of a Serebii.net entry for their page detailing the location Route 201 (Serebii.net). While Serebii.net can be very informative to a player, it can also be cumbersome to use. There is a lot of text a user has to process in order to learn what can be found in each location in a certain game. This makes Serebii.net unsuitable for new or casual players.

Another popular tool which is similar to our visualization is Bulbapedia. Bulbapedia is organized a lot like Wikipedia. Every location has a different page, which provides detailed information, about that location for the games *Pokémon Diamond*, *Pokémon Pearl*, and *Pokémon Platinum*. Each page lists each Pokémon that can be found at the corresponding location, detailing the various encounter parameters and how they change the Pokémon encountered. Like Serebii.net, Bulbapedia also does not show how the total percent is lowered to 100% after the encounter parameters are applied. Figure 2 shows a screenshot of most of the Bulbapedia page for the location Route 201 (Bulbapedia). Bulbapedia is very useful to figure out how to catch a particular Pokémon at a particular location and its less cluttered layout is friendlier to newer users. However, since Bulbapedia still organizes its information on multiple pages making it unsuited for serious players and Nuzlockers, who want to know PEDs quickly.

Morning					
Pokémon Diamond					
Starly	50%		Bidoof	50%	
Min Level	Max Level		Min Level	Max Level	
2	3		2	3	
Pokémon Pearl					
Starly	50%		Bidoof	50%	
Min Level	Max Level		Min Level	Max Level	
2	3		2	3	
Pokémon Platinum					
Starly	50%	Bidoof	40%	Kricketot	10%
Min Level	Max Level	Min Level	Max Level	Min Level	Max Level
2	3	2	3	3	3

Figure. 1: Serebii.net *Route 201* 2008 A Screenshot of Route 201 for the Sinnoh Region on Serebii.net

April 2018

Pokémon	Games	Location	Levels	Rate		
				Day	Night	Other
Staryu	D P Pt	Grass	2-3	50%		
Bidoof	D P Pt	Grass	2-3	50%		
Bidoof	D P Pt	Grass	2-4	40%	50%	40%
Krikitot	D P Pt	Grass	3	10%	0%	10%
Gift Pokémon						
Turtwig	D P Pt	Starter Pokémon	5		One	
Chimchar	D P Pt	Starter Pokémon	5		One	
Piplup	D P Pt	Starter Pokémon	5		One	
Swarm						
Doduo	D P Pt	Swarm	2		40%	
Doduo	D P Pt	Swarm	2-3		40%	
Special Pokémon						
Nidoran♀	D P Pt	Poké Radar	3		10%	
Nidoran♀	D P Pt	Poké Radar	2		2%	

Figure. 2: Bulbapedia *Route 201* 13 April 2008 A Screenshot of Route 201 for the Sinnoh Region on Bulbapedia April 2018

2.2 Academic Papers

Early on, we decided that we wanted PokeMap to be an interactable map of the entire region within *Pokémon Platinum*. Therefore, one of the first topics we wanted to find material on was the creation and usage of maps as data-conveying visualizations. The paper “Animation and the Role of Map Design in Scientific Visualization” by David DiBase, et. all worked well for this. The paper discusses how best to create a map that visualizes scientific data, drawing examples from fields like Earth Science. DiBase laid out how users absorb information from maps: a combination of visual thinking and visual communication. This helped us consider how a user might read our visualization. The most useful part of the paper for us, however, was how DiBase defined static maps and dynamics maps. Static maps are one unchanging visualization, whereas dynamic maps change according to certain dynamic variables like duration, order and rate of change. Reading the paper helped us work out how we wanted encounter parameters to affect the visualization. We decided to go with a static map, since only the PEDs change with encounter parameters and the shape of the map itself does not.

Throughout the design of our visualization, we were worried about covering too much of the map at one time. We knew that we wanted to overlay the sprites of Pokémon and the PEDs of select locations on top of the map and were cautious about making a large amount of the map hidden. We tried to find a

paper on clutter in visualizations and found the paper “The Chartjunk Debate a Close Examination of Recent Findings” by Stephen Few. This paper investigated the use of embellishments with different types of graphs, finding that graphs with a moderate amount of embellishments were more memorable and more liked by users. Unfortunately, the paper wasn’t as relevant to the issue of clutter and overlaying, since the overlays displayed new information and weren’t just embellishments of information already being displayed. The paper was also quite similar to “Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts” by Scott Bateman et al., which we read in class. We did try to include some embellishments in graphs which show the PEDs of a location, such as the sprites of individual Pokémon, to make our graphs more memorable and easier to read.

Initially, we planned to use pie charts to show the PEDs of individual locations. While we have been warned that pie charts are flawed, we thought that they would be a familiar and quick-to-read visualization to show the chances for each Pokémon to be encountered. We wanted to find a paper that would detail how to effectively use pie charts. The paper “Integration Processes during Frequency Graph Comprehension: Performance and Eye Movements While Processing Tree Maps Versus Pie Charts” by Lynn Huestegge and Tristan Potzsch helped us understand how a user reads a pie chart. The paper broke down the three steps behind reading a graph: pattern recognition, interpreting the recognized pattern, and decoding the pattern. Huestegge and Potzsch detailed how users go through these three steps with tree maps and with pie charts and listed some of the advantages and disadvantages of pie charts. A key disadvantage included difficulty comparing surface areas of slices, and the key advantages were size ordering and great familiarity. We had planned to mitigate the disadvantage by including the percentage of the whole pie that each slice took up and gain the benefit of high familiarity. Unfortunately, we didn’t use what we learned in the paper in our final visualization, since we switched from pie to bar charts after receiving feedback.

3. Design Evolution

Our design was split into two parts. The first part was the full map view, which would show the names of locations and the Pokémon available in those locations. The second part was the PED view that showed the specifics of the PED. These views would be overlaid on the map. Both of these views would take data from two prepared json files storing the encounter data and general Pokédex information.

3.1 Map View

The original plan for the map view was to have each location be represented by rectangles on the map. Each location visible from the map was then given one or more rectangles. The rectangles were given a title property so that when hovering over the image, the name of the location will appear.

Each location was to have a tooltip showing what Pokémon were available there. Initially, these tooltips were going to be on the map constantly, but we quickly decided that tooltips should only appear when the user mouses over a location. To get the images of the Pokémon we used a sprite sheet of small Pokémon sprites and inserted them onto a canvas. We determined where to find the Pokémon on the sprite sheet by using that Pokémon's Pokédex number (an id number each Pokémon has that the sprite sheet was sorted by). The issue we found is that the canvas would cover the map and we would no longer be hovering over the location. This caused flickering as the mouse switched from hovering over the location to hovering over the canvas.

The flickering issue also prevented clicking on the map to work properly so we had to choose a new way to display the sprites. Instead of using canvases, we decided to take advantage of the fact that we are using D3.js for other parts of this program and used D3 to show the sprites. Using D3, we managed to make a floating div that contained the Pokémon sprites and the location name when the mouse hovered over a location. Since the mouse was only every hovering over one thing, the flickering ceased and clicking worked normally. The floating div tag also makes the location name a bit easier to see since we are now able to edit its font.



Figure 3: First working version of the PokeMap map view with location names.

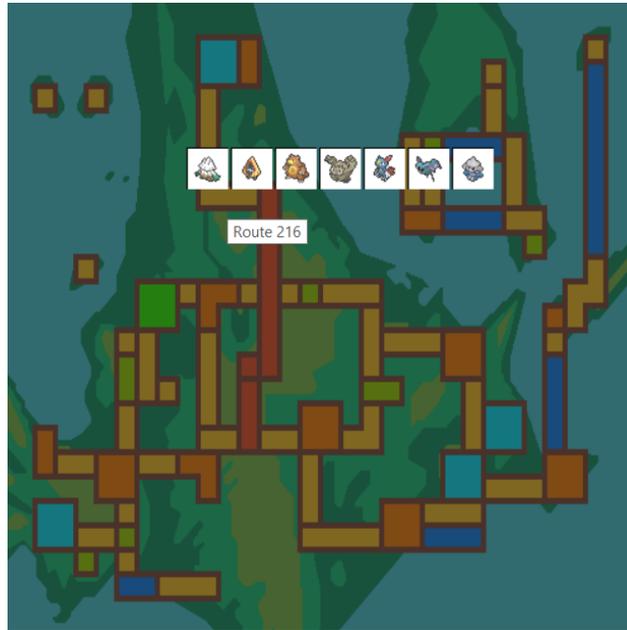


Figure 4: An early version of the PokeMap map view with the Pokémon sprites on a canvas.

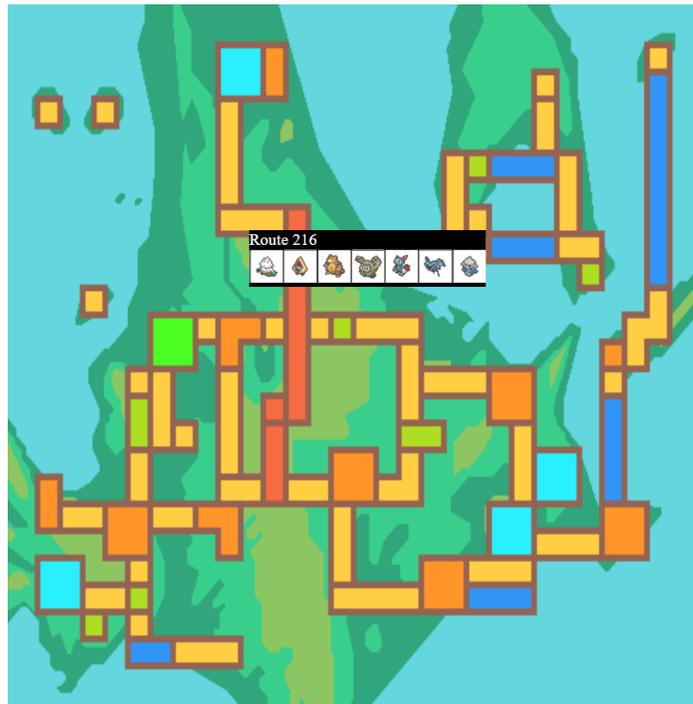


Figure 5: The current version of PokeMap's map view using D3.

3.2 PED View

We planned to use D3 to visualize the PED in this view. We started with a pie chart because it seemed like a good visualization to show a percentage distribution. We also wanted to include Pokémon images in the chart along with their names to make it easier to recognize which slice contained which Pokémon. However, when there were too many slices in the pie, images and writing from some slices were getting cut off.

After getting feedback from the class, we decided to switch from a pie chart to a bar chart instead. Switching fixed the overlapping issue and it also made it easier to compare the percentages. Since it was easier to compare, we removed the text showing the exact value. We also sorted the bars by size to make the graph easier to read. We also began to play with how the user would change the encounter parameters of the PED. We started with checkboxes primarily because they are easy to debug with. We planned to switch from them to a drop down menus for the final version because not all the encounter parameters have binary values.

The last few steps were to add the PED view to the map view when the user clicked on a location and to switch the user input from checkboxes to drop down menus. If the user clicked on any location, the PED view for that location would appear as an overlay. Double clicking off the overlay would cause the PED view for that location to be removed. We switched most of the check boxes to four drop down menus. The drop down menus divided the encounter parameters into four groups: floor/sub-area, type of encounter, time of day, and GameBoy Pak. We left PokéRadar and swarm as checkboxes since their values are binary. We also configured the PED view such that only encounter parameters that could create different PEDs would appear in the drop down menu.

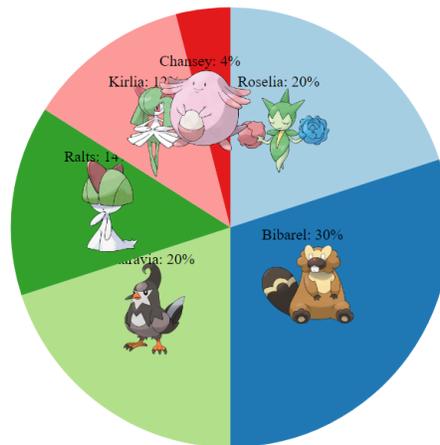


Figure 6: The initial version of the PED view using a pie chart.

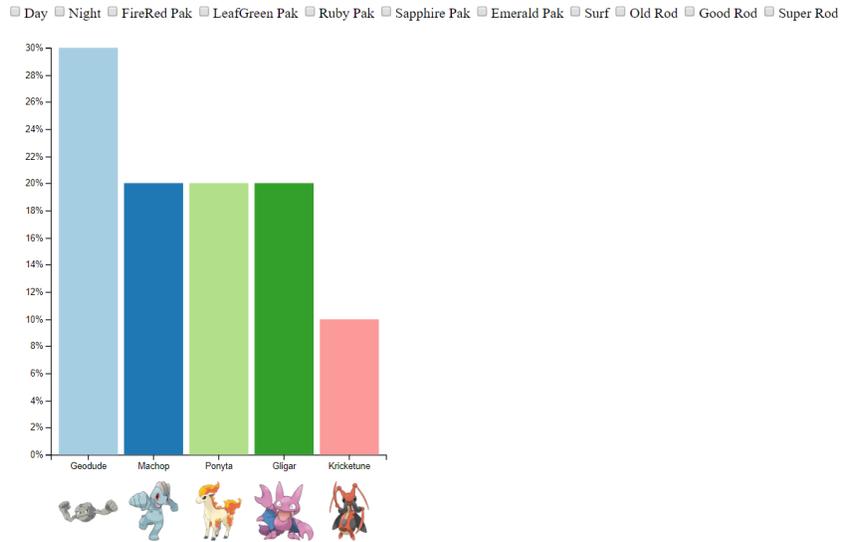


Figure 7: The PED view using a bar chart and having checkboxes for user interactivity.

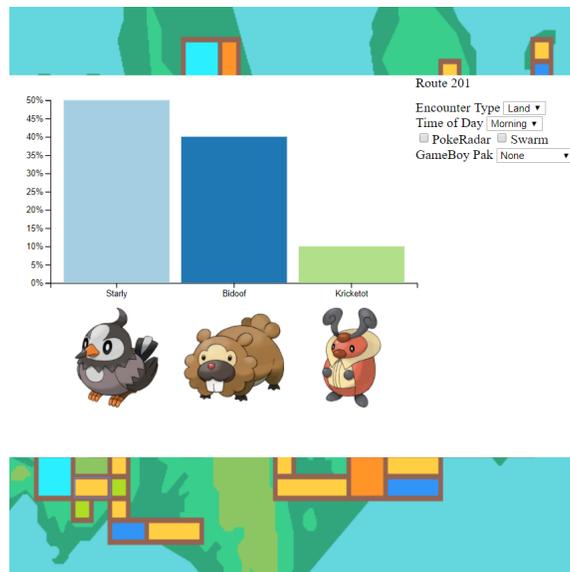


Figure 8: The current version of the PED view being overlaid over the map view.

4. Feedback

Since the project was divided into two parts, we decided to ask for feedback on both of those parts. For the map view, we asked users how we should be visualizing multi-floor/sublocation data. For the PED view, we asked how we should improve our visualization beyond the pie chart we already had.

Feedback for the map view was very mixed. Three users thought we should have the map view display all of the Pokémon that could be found in a location, and only worry about sublocations in the PED view. Three users thought we should have the tooltip display multiple rows of sprites, one row for each sublocation. We decided to only worry about sublocations in the PED view, to keep the size of the tooltip small. Since there are locations with more than 10 floors, it is very likely that dividing the tool tip into vertical sections would cause the tooltip to trail off the page.

Feedback for the PED generalized down to not using pie charts. The most common recommended alternative was to use a bar chart over using a pie chart since data would be unable to overlap itself. Bars also make it easier to compare percentage values at a glance. We were also recommended that we should probably have the bars show their exact value in a tooltip when hovered over. However, we did not have the time to implement this feature.

5. Core Features & Technical Implementation

PokeMap is written as a web app powered by the D3.js library. We acquired our data from PokemonSlots (Shiny Finder), which has the slot data for multiple Pokémon games including *Pokémon Platinum*. We would have rather have taken the slot data directly from the game, but due to time constraints, we were unable to. The slot data was converted from a .csv file to a .json file by a python script to make the data easier to parse.

Note that slot data is not the same as raw percentage data. An example of what slot data looks like can be found in Figure 9. Each slot contains one Pokémon though one Pokémon can appear in multiple slots. When a wild encounter is triggered, the games chooses one of these slots as the Pokémon the player encounters. Each slot is also weighted such that earlier slots are more likely to occur than later slots. We use these weights to convert the slot data to percentage data. We use the conversions found by PI Hudson. Whenever an encounter parameter is changed, some of the slots change the Pokémon within them. For example, on Route 203 when there is a swarm, slot 0 and slot 1 change to Cubone. This change is the specific reason we chose to use slot data over raw percentage data as raw percentage data does not work well when multiple encounter parameters change their value.

Area	0	1	2	3	4	5	6	7	8	9	10	11
Route 203	Starly	Shinx	Starly	Kricketot	Bidoof	Abra	Abra	Shinx	Starly	Bidoof	Starly	Bidoof

Figure 9: Shiny Finder *PokemonSlots* The default slot data for Route 203 on land. The percent distribution from this slot data is {Starly: 35%, Shinx: 25%, Bidoof: 15%, Abra: 15%, Kricketot: 10%}

Both the map view and the PED view take advantage of the D3.js library. For the map view, we take advantage of D3's mouse events to determine when we are hovering over or clicking on locations. The map uses the mouse events to reveal or hide various div tags that we can add desired information to. The PED view uses D3's bar related methods to create bar charts and D3's change event handler for the drop down menus and the checkboxes. Using these changes, the PED will change the slots that are being used to generate the bar chart.

6. Division of Work

We worked evenly on the written documents and on the presentation. We both decided to tackle one of the two views of our project to be put together at the end.

6.1 Jared Okun

I focused on the data parsing and the PED view. I parsed all the slot data into a json file with a python script. I also reparsed the data a few times when we needed to change how the data looked for sub locations. For the PED view, I created the script to show the pie/bar chart in its own web app until I fused it with the map view. I was the one who fused the two views. I also did a majority of the bug fixes related to D3 because I have more experience with the library than John does because of Submittity.

6.2 John Hulton

I focused on creating the map view. I found where every location is relative to the whole map so that the map could be fit to the screen without losing track of the locations. I then used HTML, javascript and css to create an element for each area and wrote mousemove and mouseout functions such that mousing over an area would show the sprites of every Pokémon that could be caught in that location. This required writing a script to retrieve information from two json files and a png file while managing overlays (which Jared helped me with). I also did most of the academic research.

6. References

David DiBase, et. all "Animation and the Role of Map Design in Scientific Visualization." 1992
Cartography and Geographic Information Systems, Vol. 19, No. 4, 1992, pp. 201-214, 265-66

Fanzezi "Pokemon-DB" 18 July 2016 <https://github.com/fanzezi/Pokemon-DB>

Few, Stephen “The Chartjunk Debate A Close Examination of Recent Findings” Perceptual Edge Visual Business Intelligence Newsletter 2011 perceptualedge.com

Huestegge, Lynn and Tristan Herbert Potzsch. “Integration Processes during Frequency Graph Comprehension: Performance and Eye Movements While Processing Tree Maps Versus Pie Charts.” Applied Cognitive Psychology, no. 2, 2018, p. 200

PI Hudson “View Full Version : Research: [Documen]Pokemon encounter ratio in Adv-map” The PokéCommunity Forums, 1 June 2009. PokéCommunity
<https://www.pokecommunity.com/archive/index.php/t-180793.html> Accessed: 24 April 2018

“Route 201” Serebii.net 2008 <https://ww.serebii.net/pokearth/sinnoh/route201.shtml> Accessed: 24 April 2018

Shiny Finder “PokemonSlots” <https://sites.google.com/site/pokemonslots/> Accessed: 24 April 2018

“Sinnoh Route 201” Bulbapedia April 2008 13
https://bulbapedia.bulbagarden.net/wiki/Sinnoh_Route_201 Accessed: 24 April 2018