

# Pokémon GO Analyser

Last summer, Pokémon GO was all the rage (and for some it still is). If you have a smartphone, there is a good chance you have played it yourself. For the people who haven't, we give a short summary of the aspects of the game that are relevant for this assignment: in Pokémon GO, little fantasy creatures called Pokémon randomly spawn (appear) at different times at different places in the real world. Players have to roam the world with their smartphone in hand to encounter wild Pokémon. Upon encountering a Pokémon, players throw Pokéballs (by swiping their finger) in order to catch the Pokémon. One of the main goals of this game is to *catch 'em all*. The game has many more aspects, e.g., fighting between Pokémon, ..., but they are not relevant to this assignment.

This summer, from Monday 5/9 to Sunday 11/9, the Professor (Professor Oak, the character from the Pokémon animated TV series, not Professor Schrijvers or Janssens) tracked the locations and times of the Pokémon that have spawned at this campus. Now, the Professor has chosen you to analyse this data to see whether we can derive some interesting patterns. We will not pretend to derive anything statistically significant from the data, we're just going to try to answer some basic questions like: which Pokémon spawns most often? At which spawn point do the most Pidgeys spawn? On which day of the week do the most Pokémon spawn? How many Pokémon spawn during the day, how many during the night?

## Assignment

- Implement the functions from the assignments below.
- Skeletons of the functions are given in the template.  
**You are not allowed to change their type signatures, but you can put arguments before the =.**
- Read the template to learn more about the data types used in the functions.
- After assignment 2 you can **choose the order** in which you complete the remaining assignments.
- You are allowed and encouraged to write helper functions, data types, type classes, etc.
- Moreover, you will notice that the code of some assignments looks very similar to that of other assignments. **DO NOT copy paste that code, but write higher-order functions that you can reuse.**
- **For readability, variables from the template are used in the examples. So the output you get will not literally match that of the examples.**

## 1 Projection functions

```
spawnPokemon :: Spawn -> Pokemon
spawnLocation :: Spawn -> Location
spawnTime     :: Spawn -> Time
```

Implement the three functions above that project a `Spawn` on one of its three parameters.

## 2 Group a list of Spawns by a given function.

```
groupSpawnsBy :: Eq k => (Spawn -> k) -> [Spawn] -> [(k, [Spawn])]
```

This is a helper function that should make writing the other assignments much easier.

For each `Spawn` in the list, apply the function to get a `k`. Make a list of key-value-pairs where the keys are all the different `ks`, and the values are lists of `Spawns` that have the same `k`.

```
>>> groupSpawnsBy spawnPokemon testSpawns
[("Pidgey", [pidgeySpawn1, pidgeySpawn2]), ("Pikachu", [pikachuSpawn])]
```

## 3 Which Pokémon spawns most often?

```
mostCommonPokemon :: [Spawn] -> [(Pokemon, Int)]
```

Return a list of `(pokemon, n)` pairs where `n` is the number of times the `pokemon` has spawned. This list is sorted by `n`: the pair with the highest `n` comes first and the pair with the lowest `n` comes last.

Hint: have a look at the `sortBy` function in the `Data.List` module.

```
>>> mostCommonPokemon testSpawns
[("Pidgey", 2), ("Pikachu", 1)]
```

## 4 At which spawn point does the given Pokémon spawn most often?

```
topSpawnPointsOf :: Pokemon -> [Spawn] -> [(Location, Int)]
```

Return a list of `(location, n)` pairs where `n` is the number of times the given Pokémon has spawned at `location`. This list is sorted by `n`: the pair with the highest `n` comes first and the pair with the lowest `n` comes last.

```
>>> topSpawnPointsOf "Pikachu" testSpawns
[((50.864605, 4.6786203), 1)]
```

## 5 During which hours do the most Pokémon spawn?

```
topHours :: [Spawn] -> [(Int, Int)]
```

Return a list of `(hour, n)` pairs where `n` is the number of times a Pokémon has spawned between `[hour:00, hour:59]`. This list is sorted by `n`: the pair with the highest `n` comes first and the pair with the lowest `n` comes last.

```
>>> topHours testSpawns
[(12, 2), (3, 1)]
```

## 6 On which day of the week do the most Pokémon spawn?

```
topWeekDays :: [Spawn] -> [(WeekDay, Int)]
```

Return a list of (weekday, n) pairs where n is the number of times a Pokémon has spawned on weekday. This list is sorted by n: the pair with the highest n comes first and the pair with the lowest n comes last.

```
>>> topWeekDays testSpawns
[(Friday,2),(Tuesday,1)]
```

## 7 How many Pokémon spawn during the day, how many during the night?

```
dayAndNight :: [Spawn] -> (Int, Int)
```

Return a tuple with as first element the number of Pokémon spawning during the day, and as second element the number of Pokémon spawning during the night. Day time is [7:00, 21:00] (both bounds included).

```
>>> dayAndNight testSpawns
(2,1)
```

## 8 How many Pokémon spawn around the hour and how many between the hours?

```
aroundTheHours :: [Spawn] -> (Int, Int)
```

Return a tuple with as first element the number of Pokémon spawning around the hour, and as second element the number of Pokémon spawning between the hours. With “around the hour” we mean from 15 minutes (including 45) before the hour to 15 minutes after the hour (excluding 15), or: [(x - 1):45, x:15[. All other times are “between the hours”, e.g., 12:30, 7:43, 18:16.

```
>>> aroundTheHours testSpawns
(2,1)
```

## 9 Analyse the spawn data.

```
analyseSpawns :: IO ()
```

Load the [Spawn] from spawns.data. The spawns.data file contains spawn data which you can convert from a String to [Spawn] using the read method of the Read type class.

Hint: have a look at the readFile function in the System.IO module.

Analyse this data using the functions you have written and print the results using IO. The output should look like this:

```
>>> analyseSpawns
# Most common Pokemon:
1. Pidgey spawned 524 times
2. Rattata spawned 491 times
3. Weedle spawned 484 times

# Top spawn points of Pidgey:
1. at (50.863804,4.6802573) 40 Pidgeys spawned
2. at (50.864193,4.6786504) 38 Pidgeys spawned
```

3. at (50.86296,4.674903) 38 Pidgeys spawned

# Top hours:

1. at 20 o'clock 180 Pokemon spawned
2. at 16 o'clock 180 Pokemon spawned
3. at 15 o'clock 180 Pokemon spawned

# Top week days:

1. on Wednesday 720 Pokemon spawned
2. on Saturday 719 Pokemon spawned
3. on Friday 719 Pokemon spawned

More Pokemon spawn during the day than during the night: 2511 vs 1588

More Pokemon spawn around the hours than between the hours: 2596 vs 1503

**N.B.** the top spawn points of the most common Pokémon according to the data are shown. Pidgey is **not** hardcoded.

When more Pokémon spawn during the night than the day, you can swap the words “day” and “night” so it reads: “More Pokemon spawn during the night than during the day: x vs y”. Do the same for around and between the hours.

When the number of Pokémon spawning during the day and the night is the same, you can print “As many Pokemon spawn during the day as during the night: x”, where “x” is the number. Similarly for around and between the hours: “As many Pokemon spawn around the hours as between the hours: x”.

In case of an ex-aequo, e.g., “Top hours”, the order does not matter.