Assignment 3
DEV 2 - Year 2015-2016
HOF's

The Dev TEAM

January 4, 2016

1 Goal and description

The goal is to improve your design and implementation skills about *HOF's* (high order functions). For this assignment you may your code form Assignment 2.

2 Software requirements

To work with the simulation you need `PyGame 3.4` and `Python 3.4`. You can download the `PyGame 3.4 x86` from https://bitbucket.org/pygame/pygame/downloads/pygame-1.9.2a0-hg_ea3b3bb8714a.win32-py3.4.msi. PyGame is a set of Python modules designed for writing games. The simulation comes with a *template project*. The template is available on N@school and GitHub under the voice `Assignment 3`.

3 Details

HOF's - revision In the following we show the HOF's necessary to solve this assignment: `map` and `filter`.

map is a high order function that takes as parameters a list `l` and a transformation function `f` and returns a *new* list made of all elements of `l` transformed by `f`. In order to use the `map` function the transformation function `f` **must** always return the transformed value. In the following example elements of a list of numbers are all increased by one:

```
numbers = Node(1, Node(2, Node(3, Empty)))
updated_numbers = map(numbers, lambda n: n + 1)
```

filter is a high order function that takes as parameters a list `l` and a predicate function `p` (which evaluates a boolean condition) and returns a new list made of only the elements of `l` that satisfy the predicate `p` (for which the condition return `True`). In order to use the `filter` function the predicate `p` must always return a boolean value. In the following example we select the even numbers from a list of numbers:

```
numbers = Node(1, Node(2, Node(3, Empty)))
filtered_numbers = filter(numbers, lambda n: n % 2 == 0)
```

iterate is a high order function that takes as parameters a list `l` and a predicate `p` and applies the predicate to each element of the list `l`. Note that the function returns nothing, which means that we just "do" something "with" each element, ideally leaving the original list intact. In the following example we print every element of a list:

```
numbers = Node(1, Node(2, Node(3, Empty)))
iterate(numbers, lambda n: print(n))
```

4 Tasks

Task 1 [**HOF**] adapt your code from the Assignment 2 so to include HOF's. Precisely, your HOF's should replace your previous *draw all cars/draw all boats* and *update all boats/update all cars*. Design and implement such code.

Hint 1 For drawing your boats and cars use `iterate`.
Hint 2 The function `iterate` is not provided. Try to design and implement it.
Hint 3 For moving boats and cars use `map`.
Hint 4 For removing boats and cars that reached their destination use `filter`.

5 Submission and deadline

Contribution: *Groups of 2 students is allowed with individual responsibility*
What: *One PDF per student for all code + comments (comments: explain your code)*
When: *The Friday of week 7*
Where: *On N@school*

GOOD LUCK!!! The Dev TEAM ☺