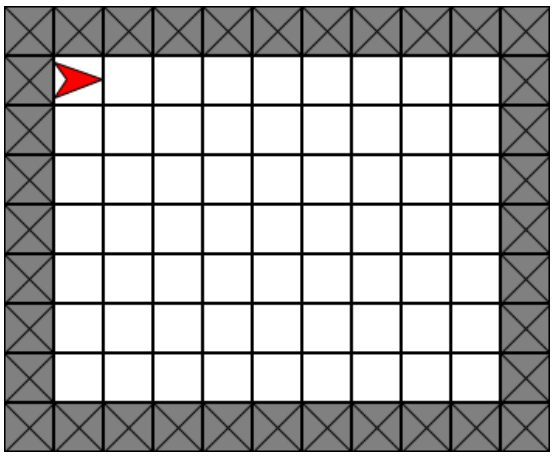
# Bonus assignment 2: Journey to the center of the earth

Course 'Imperative Programming' (IPC031)

## 1 Assignment

The goal of this assignment is to equip Charles with an agent that will make Charles find the center of his world. The starting situation is an empty world, surrounded by walls. You are allowed to assume that there is an odd number of lanes as well as an odd number of streets, that need not have the same value. An example is shown below:

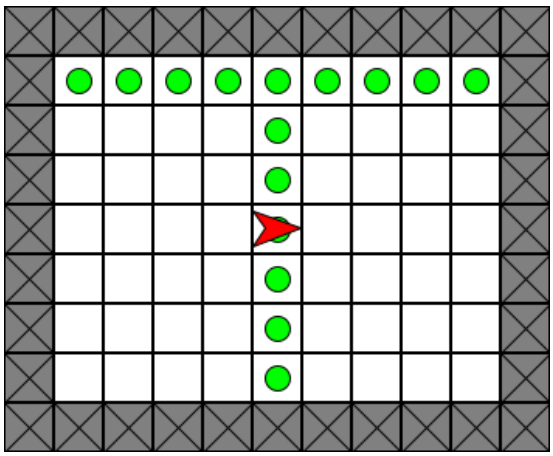


This problem can be solved in two steps. To obtain a "sufficient" grade for this bonus assignment, it is okay if you succeed in solving the first step (Part 1). To obtain "good" grade, you should solve the second step as well (Part 2).
In this assignment you use only the following C++ language constructs and Charles constructs that have been explained in the lecture:

- C++: sequence (`;`), choice (`if-else`), conditional repetition (`while`), logical operations; functions;
- Charles: `step`, `turn_left`, `turn_right`, `get_ball`, `put_ball`, `in_front_of_wall`, `north`, `on_ball`;
- functions to structure your code.

### Part 1: Find the center

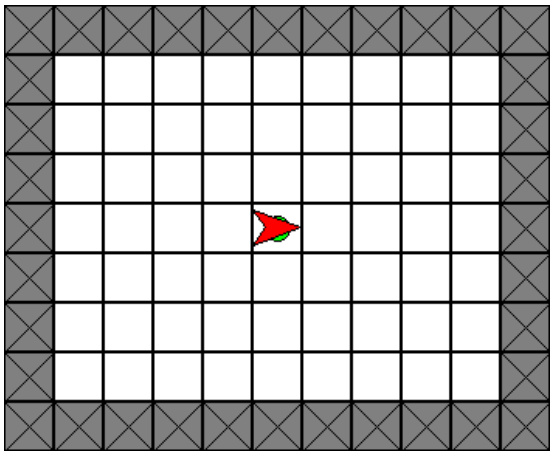
A possible strategy to find the center of the earth is to keep placing balls at opposing walls of Charles world, until they meet. At that position, you have found the middle of one the axes of Charles world. This is illustrated in the figure below:



Adjust the function `find_center_agent ()` in "`agent.cpp`" to implement this strategy. This function can be invoked by the Charles command "Bonus: find the center...".

### Part 2: Clean up balls

Assuming that Charles is standing at the center of the earth, we need Charles to remove all balls that are not at the center of the world. At the end, Charles needs to stand on the ball at the center of the world, facing East. This is illustrated in the figure below.



Adjust the function `clean_up_agent ()` in "`agent.cpp`" to implement this part. This function can be invoked by the Charles command "Bonus: clean up balls. . . "'.

## 2 Products

As product-to-deliver you only need to upload to Brightspace the "`agent.cpp`" file that you have extended with solutions for each problem-solving-algorithm.

## Deadline

**Bonus assignment:** Monday, September 18, 2023, 15:30h