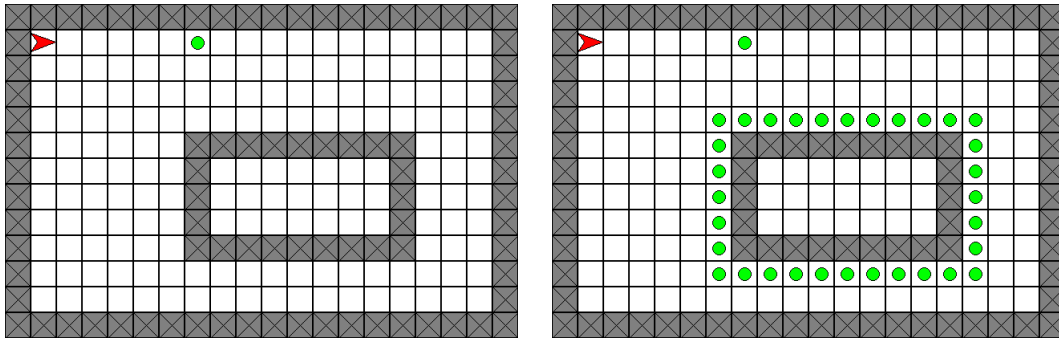# Bonus assignment week 1–2: Advanced Charles

Course 'Imperative Programming' (IPC031)
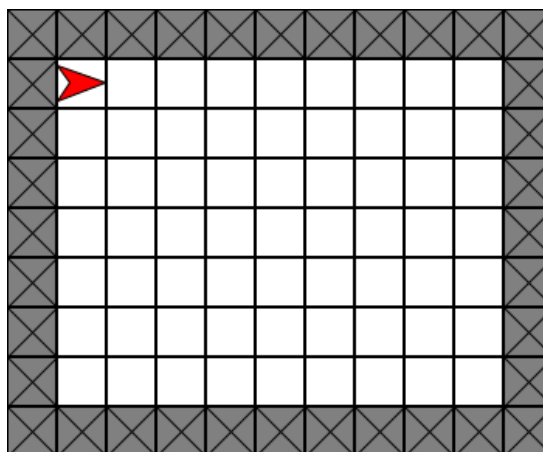
1 Assignment

Part 1: Around the block

In "`agent.cpp`", the function `block_agent` () creates a wall structure in the middle of Charles' world (left image). This function can be invoked by the Charles command "Bonus: Around the block...".

It is your task to make Charles walk around the wall structure in the middle of his world, while dropping balls right next to the walls, and finally return to its starting position (right image). Your solution must work for any rectangular wall structure but different size. Charles can find the 'north-west' corner of the rectangle via the ball that is placed on the 'northern-most' lane. Solve the problem using only Charles' facilities and the core language elements that have been discussed in the lecture.

Part 2: Journey to the center of the earth

The goal of this part of the 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:
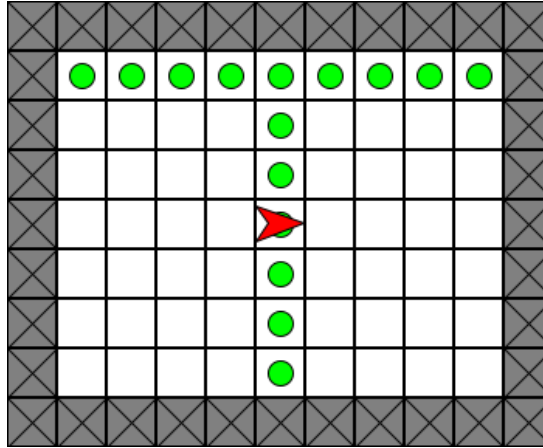

This problem can be solved in two steps. To obtain a "sufficient" grade for this bonus assignment, it is okay if you succeed in solving the first step (Part 2.1). To obtain "good" grade, you should solve the second step as well (Part 2.2).

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.
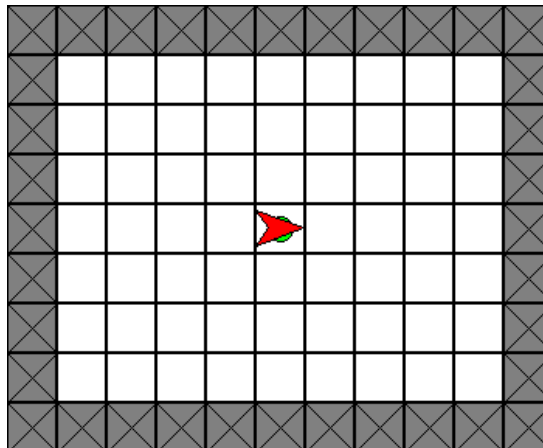
Part 2.1: Find the center

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 2.2: 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: Friday, September 13, 2024, 23:59h

Important notes:

1. check that you have actually submitted your solution in Brightspace.
2. the deadline is firm, and it is impossible to upload a solution after expiry. In that case you fail the assignment.
3. you can upload solutions to Brightspace several times, but only the last submission is stored.
4. identify yourself and your lab partner in every uploaded document. The identification consists of your first and last names, student numbers, and number of (sub) assignment. By identifying yourself, you declare that the uploaded work has been created solely by you and your lab partner.
5. your work will be judged and commented upon. We expect that you obtain the feedback, read it, and use it to for the next exercises.
6. it is essential that you only submit your own solution, never copy somebody/something else's solution, and never share your solution—in particular: **AI tools (including but not limited to Github Copilot or ChatGPT) are not permitted**, solutions from previous year cannot be reused, and finally, you and your lab partner take joint responsibility for the assignment you submit.