Worksheet 4.4 / Cleaning up 1

Ceebot environment:
"SB 4: Variables" – "Cleaning up 1a"

Mission:
The five empty batteries have to be destroyed.

Concept:
The robot has to
- locate one battery
- turn in its direction
- fire its gun

and repeat these steps _____ times.

New commands:
- The command `fire(…)` causes the robot to shoot a volley. The duration of the volley (in seconds) is determined by the command's parameter.
- The command `aim(…)` adjusts the gun of the robot vertically. The parameter determines the angle of the gun relative to the horizontal. Use the parameter `-5` to fire at the batteries which are in a distance of 10 meters.
- The function `direction_to(…)` returns the horizontal angle towards the nearest object of a certain category found **in front of the robot**. The category is determined by the command's parameter.

Listing 4.1.1 – Aufräumen 1a

Program code	Description
`extern void object::Shooter()`	
`{`	
`  aim(-5);`	
`  _____ angle;`	_____ Declaration of the variable `angle`, which stores a decimal number
`  _____`	Repeat _____ times
`  {`	
`    angle=direction_to(_____);`	_____
`    turn(____);`	Turn towards the battery
`    fire(1);`	_____
`  }`	End of repeated commands
`}`	

Further exercises
1. In "SB 4: Variables" – "Cleaning up 1b", the robot has to destroy another set of five empty batteries which are further away. Thus, the robot has to move up to a distance of 10 meters in order to fire a volley.
2. In "SB 4: Variables " – „Drawing board", the robot should be programmed to follow the astronaut (use the category `Me` as parameter for `direction_to` und `distance_to`) by permanently turning towards the astronaut and moving up a bit less than the distance towards the astronaut. Use a `repeat(1000)` – loop for frequent repetition of a set of commands .

Worksheet 4.5 / Stockyard 1

Ceebot environment
”SB 4: Variables“ –”Stockyard 1“

Mission
The robot has to place the six batteries at the stockyard in a row, so that there is a distance of one meter between the batteries.

Concept
In order to place the batteries correctly, the robot has to
- determine the distance to the next battery and memorize this number,
- determine the direction to the next battery and memorize this number,
- move to the battery and grab it,
- drive back to the starting position,
- turn towards the starting orientation,
- move a certain distance between the barriers (about 10 meters for the first battery, and one meter less for each subsequent battery)
- drop the battery behind the robot and
- move back to the starting position.

All batteries are collected when these instructions are repeated _____ times.

The robot needs to memorize the following numbers:

- The distance to the nearest battery, which is stored in the variable `dist_PowerCell`, needs to be memorized because _____
- The direction to the nearest battery, which is stored in the variable `dir_PowerCell`, needs to be memorized because _____
- Finally, the variable `dist_Stockyard` is used to memorize the distance the robot has to drive into the alley between the barriers.

Remarks
- When the variable `dist_PowerCell` determines the distance to the next battery and the robot executes `move(dist_PowerCell)`, the robot drives a bit too far. The robot needs to stop 1.5m **in front of** the battery and grab it.
- In order to place the batteries in a row, the distance `dist_Stockyard` has to be changed with each collected battery!
- Use `float` variables to save the distances and directions. `int`-variables are not practicable because

  _____
- Why is it important that `direction_to` und `distance_to` only detect objects in front of the robot?

  _____

Use comments to improve your program’s “readabilty“, such as `// grab next battery`.