Advance Topics In Programming – Exercise 1
Design explanation

Submitted by: Noa Leibovich (305220980) Sharon Naftaly (300689775)

We will describe here the design of our simulator program. We have made an attempt to make our solution suitable for the next exercises. Hence we will describe here a solution that can handle several houses and algorithms, although specific implementation changes may be needed to meet specific requirements.

First we have a main function which is responsible for checking the input.

The main function initializes a Simulator object, who handles running the simulation:
- The Simulator c'tor gets the paths to the configuration file directory and houses directory and uses the ConfigManager and HouseManager objects to store the information:
    - The ConfigManager reads and store the configuration read from the config file.
    - The HouseManager reads the simple1.house file and stores a House objects list of the houses read.
- The simulator c'tor initializes the algorithms list, currently with one SimpleAlgorithm object (the SimpleAlgorithmclass implements AbstractAlgorithm)
- The simulator holds a list of AlgorithmRunner - on object for for each algorithm.
    - Each AlgorithmRunner in the listholds a pointer to its algorithm and a sensor object which is passed to the algorithm. The sensor object implements AbstractSensor, and holds a pointer to the current house, and the current robot location in the house.
    - The AlgorithmRunner object handles the run of each algorithm on the current house, and saves needed info about the algorithm run on the house for score calculation, such as number of steps done by the algorithm, and dirt collected, as well as the algorithm finishing state for the current round.

In case opening/reading the files fails, the simulator will set the initSuccessful flag to false. The main function then checks it and in case it is False, the program will close with return code 1, with the appropriate error message.

The main function then calls the runSimulation function of the simulator:
- This function goes on every house, and for each house, while there are algorithms that are still running and steps remaining, will run each algorithm on the house for one step at a time (round-robin).
- The AlgorithmRunner object of each algorithm stores a copy of the current house in its house data member, and the sensor is updated upon house change.
- In each simulation step for the current house - the algorithm queries the algorithm runner for the state of the run - in case it finished successfully, has no more battery or made an

invalid step, the algorithm runner will stop the simulation for its algorithm by marking a suiltable flag in its SimulationFinishState data member.

- When the simulation is done for a specific house, each AlgorithmRunner calculates the score for its algorithm and places it in a score list (ordered as the houses list). On the next house, the algorithmRunner configurations are reset.

Once the simulation is over the, simulator prints the score list of each algorithm. Currently it will print on number - for one algorithm with a score list for one house.