Assignment 3

TPK4186

Torstein Heltne Hovde
Lars Magnus Johnsen
Simen Eger Heggelund

Intro:
In this assignment we are simulating the operation of a warehouse, where the stockage and the destockage of products is realized by autonomous robots. The program constructs a warehouse where the size is decided by the user. The program also constructs a catalog with n products, the number of products in the catalog is limited by the warehouse size, since a shelf only can contain one type of products. The simulator also constructs random deliveries and client orders, to simulate the warehouse operation with n robots.

How to run:
Assuming that most common third party libraries are installed.

1. **Pip install tabulate**. A package to format the layout of the warehouse when representing the simulation of the warehouse.
2. Run **simulator.py**. This file illustrates the creation of all the data models in its respective place within the warehouse and prints out information of all the data.
3. Run **simulator_visualize.py**. This file summarizes the whole operation of loading the warehouse with products from the deliveries with robots and the operation of retrieving products from the client orders. It is all visualized through a custom made printer that displays the warehouse second for second (0.1 for efficiency sake) during these processes with the robots in full operation.
4. Run **optimizer.py**. This file contains functions to simulate different warehouse parameters such as size and amount of robots to produce statistics for optimization. With this insight we can modify our warehouse for more seamless and effective operation. This can be automated but we thought it was handy to get presented all the statistics for the user to do qualified decisions himself.

Task 1:
The classes “Catalog”, “Cell”, “ClientOrder”, “Product”, “Robot”, “Shelf”, “Truck” and “Warehouse” manages the entites involved in the warehouse operation, and hence the simulation process.

Task 2:
In the file “warehouse.py” there are several methods to create entites. For example the functions “constructCatalog”, “constructRandomDelivery and “constructRandomClientOrder”. The function “constructWarehouseLayout” designs the warehouse from the desired size, in other words number of cells in X and Y direction.

Task 3:
The class “Printer” is within the “printer.py” file, it is responsible for printing among other things the warehouse layout and catalog. Some of the functionality of the printer class has been adopted by the simulator.py file.

Task 4:
When runned the file “simulator.py” utilizes the Printer class to print the warehouse layout and a corresponding catalog. It also reports and prints out a delivery. The number of robots is chosen by the user, and for every robot the program also prints out the assigned product and calculated route to the storage cell as well as the route “home”.

The file simulator_visualize simulates and visualizes the warehouse being filled with deliveries and also emitted by clients orders. Before running the user can choose to print both the visualization of the warehouse being filled with products and the continuously updated routes for each robot. A movie of the visualization is also uploaded together with the code.

Task 5:
The file “optimizer.py” allows the user to choose the number of robots and warehouses used in the experimental protocol. When running the file, the Optimizer class constructs 50 random deliveries for each warehouse and calculates the average time it takes to empty the delivery queue into the warehouse shelves(load). This is then plotted, to visualize the relationship between the size of the warehouse and elapsed time used to handle a delivery. This data can be helpful for Irinoco in deciding the size of the warehouse and number of robots needed for the most effective and seamless automated warehouse operations.