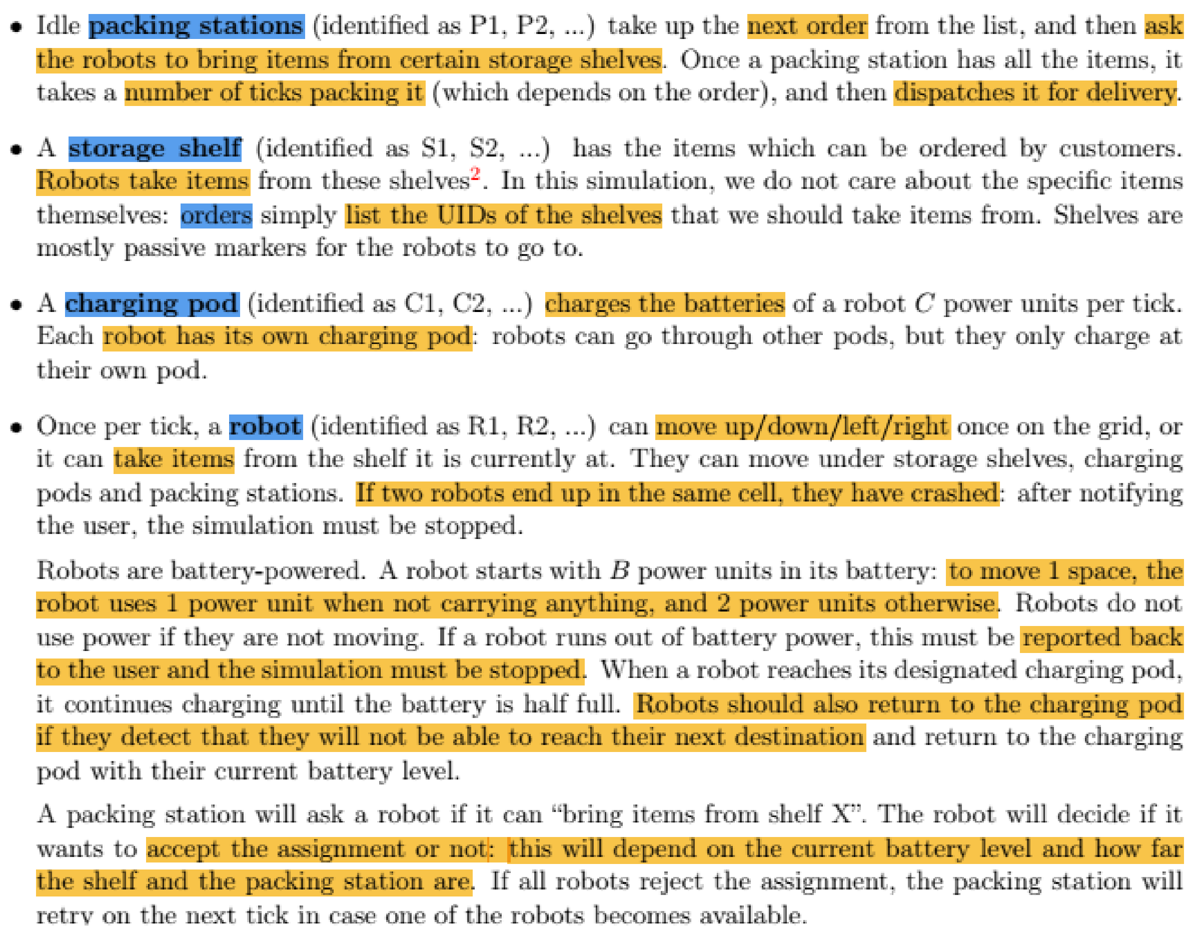
Noun-Verb analysis

Blue = noun

Yellow = verb

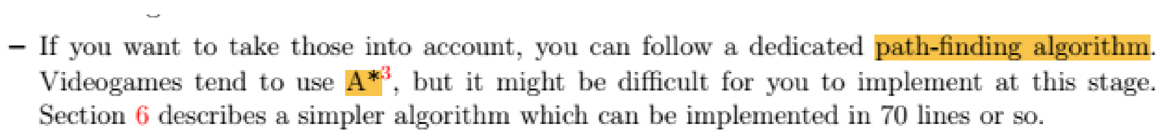
Below, are the screenshots taken from the coursework specification. Nouns and verbs have been highlighted to show classes and methods to be created.



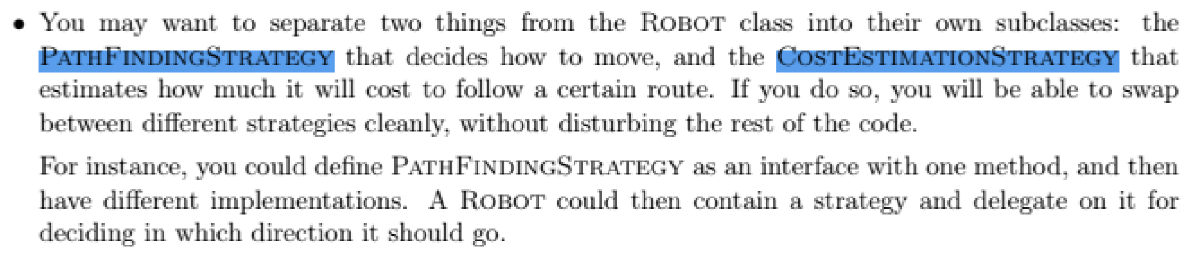
- Idle packing stations (identified as P1, P2, ...) take up the next order from the list, and then ask the robots to bring items from certain storage shelves. Once a packing station has all the items, it takes a number of ticks packing it (which depends on the order), and then dispatches it for delivery.
- A storage shelf (identified as S1, S2, ...) has the items which can be ordered by customers. Robots take items from these shelves[2]. In this simulation, we do not care about the specific items themselves: orders simply list the UIDs of the shelves that we should take items from. Shelves are mostly passive markers for the robots to go to.
- A charging pod (identified as C1, C2, ...) charges the batteries of a robot C power units per tick. Each robot has its own charging pod: robots can go through other pods, but they only charge at their own pod.
- Once per tick, a robot (identified as R1, R2, ...) can move up/down/left/right once on the grid, or it can take items from the shelf it is currently at. They can move under storage shelves, charging pods and packing stations. If two robots end up in the same cell, they have crashed: after notifying the user, the simulation must be stopped.

  Robots are battery-powered. A robot starts with B power units in its battery: to move 1 space, the robot uses 1 power unit when not carrying anything, and 2 power units otherwise. Robots do not use power if they are not moving. If a robot runs out of battery power, this must be reported back to the user and the simulation must be stopped. When a robot reaches its designated charging pod, it continues charging until the battery is half full. Robots should also return to the charging pod if they detect that they will not be able to reach their next destination and return to the charging pod with their current battery level.

  A packing station will ask a robot if it can "bring items from shelf X". The robot will decide if it wants to accept the assignment or not: this will depend on the current battery level and how far the shelf and the packing station are. If all robots reject the assignment, the packing station will retry on the next tick in case one of the robots becomes available.



– If you want to take those into account, you can follow a dedicated path-finding algorithm. Videogames tend to use A*[3], but it might be difficult for you to implement at this stage. Section 6 describes a simpler algorithm which can be implemented in 70 lines or so.



- You may want to separate two things from the ROBOT class into their own subclasses: the PATHFINDINGSTRATEGY that decides how to move, and the COSTESTIMATIONSTRATEGY that estimates how much it will cost to follow a certain route. If you do so, you will be able to swap between different strategies cleanly, without disturbing the rest of the code.

  For instance, you could define PATHFINDINGSTRATEGY as an interface with one method, and then have different implementations. A ROBOT could then contain a strategy and delegate on it for deciding in which direction it should go.

Executive summary

Each entity will generate its own unique identifier as well as outputting it;

An Item packing station will receive the next order and ask a robot to bring it from its respective shelf. An Item from an order, will be dispatched. A notification is sent to inform of an order being complete and if the packing station is occupied.

A storage shelf, involves a robot taking an item from its shelf.

An order, will list each UID of each shelf where an item can be taken from.

A charging pod, will charge each battery of each robot. A robot will have its own charging pod.

A robot, can move up, down, left and right; take an item from a storage shelf; move under an entity but cannot be under a cell occupied by another robot. When a robot is not carrying an item, a loss of one power unit occurs, if it is carrying an item, it will lose two. If a robot cannot reach its destination, it must return to its own charging pod. A robot will accept an order, dependent on its current battery level and how far the storage shelf & packing station are.

Path finding finds the shortest route via A* algorithm. A robot cannot collide into another robot.

Cost Estimation Strategy will find the optimal cost-effective route through use of A* algorithm. compares its location and the location of the robot.

Class name
Robot
PackingStation
StorageShelf
ChargingPod
PathFinding
CostEstimationStrategy
Order