

===

High Level Structure

===

Simulation

Can run a tick

Fail & Notify user

Report number of ticks & final result

WarehouseStatistics

Contains how long it took to dispatch each order (Is this how long it took the order from packing station to dispatch or from simulation start to dispatch?)

WarehouseSimulation

Contains floor

Randomly generate simulation

Read simulation instructions from file (*The Warehouse should provide a method, but the actual file reading should be done in a generic class that is not a domain class*)

Contains a list of Orders awaiting a Packing Station <- Probably do this with JCF Queue

Should keep track of outstanding orders (Just a counter should be enough)

Floor

Divided into cells <- Probably 2D Array

Add/Remove entities (Actors)

Actors stored in separate lists by type

Cell

Contains a list of entities (Actors)

Should be able to add/remove entity

Should be able to state what is in it, or return if it contains <T>

FuelCalculator

Calculate fuel required for a journey (*Note: Need to consider the ability to make it back to a charging station after order too*)

PathFinder

Find path

===

Actor

===

Actor

Can perform behaviour

Has UID

Has current location

PackingStation

Takes order

Has Current Order

Asks robots to get outstanding item(s) using list of Robots on Floor

Packs order (Once it has all items)

Dispatches for delivery – inform WarehouseSimulation that it's completed

StorageShelf

Passive marker for Robots

ChargingPod

Charges a robot's battery by C power units

Robot

Consider an order from a PackingStation (Needs to calculate path & fuel cost of orders to make this decision utilising reusable class defined earlier) and then reject or accept

Moves (up/down/left/right – should be based on calculated path)

Needs to know it's current path for moving – can use JCF queue and take the next item for its move.

Takes items from shelf it is currently at

Can move through other entities except robots - if it ends up in the same space as another robot it has crashed

Has Battery with B power units

Uses a certain amount of power units to move

Can only carry items from one shelf at a time

===

Order

===

Order

Contains the UIDS of shelves -> Probably use an Array. The list is predefined, so we know the values won't change.

Notable Design Decisions

- Separating Simulator and WarehouseSimulator into separate entities. This allows us to create a set of behaviours in Simulator that we could apply elsewhere i.e. a wildlife or city planning simulator. Then within WarehouseSimulator we can implement the shared behaviour and Warehouse specific behaviours.
- Notifying/failing handled by simulator. Specific failures for Robot crash or battery drain will be passed to it
- Opted to originally have separate lists for Orders based on status rather than a property – this means that we don't have to navigate and question the status for our next order or have a counter to know the next order. PackingStations can just take the next item off the queue for unassigned orders. After further refinement decided that assigned orders and dispatched orders weren't needed. We don't need this information, just the tick statistics that we can log against our WarehouseStatistics for how long orders taken and if there are outstanding orders.
- Placed UID against Actor as all actors share this