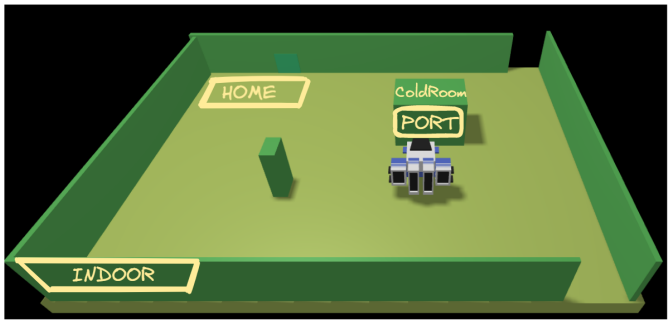
TemaFinale23

A company intends to build a *ColdStorageService*, composed of a set of elements:

1. a service area (rectangular, flat) that includes:
    - an INDOOR port, to enter food (fruits, vegetables, etc.)
    - a *ColdRoom* container, devoted to store food, upto **MAXW** kg .

    The *ColdRoom* is positioned within the service area, as shown in the following picture:



2. a DDR robot working as a *transport trolley*, that is intially situated in its HOME location. The transport trolley has the form of a square of side length **RD**.

    The *transport trolley* is used to perform a *deposit action* that consists in the following phases:

    1. pick up a food-load from a *Fridge truck* located on the INDOOR
    2. go from the *INDOOR* to the PORT of the *ColdRoom*
    3. deposit the food-load in the *ColdRoom*

3. a *ServiceAcessGUI* that allows an human being to see the current current weigth of the material stored in the ColdRoom and to send to the *ColdStorageService* a request to store new **FW** kg of food. If the request is accepted, the services return a ticket that expires after a prefixed amount of time (**TICKETTIME** secs) and provides a field to enter the ticket number when a *Fridge truck* is at the INDOOR of the service.

4. a *ServiceStatusGUI* that allows a *Service-manager* (an human being) to supervises the state of the service.

Alarm requirements

The system includes a a *Sonar* and a *Led* connected to a RaspnerryPi.

The *Sonar* is used as an ‘alarm device’: when it measures a distance less that a prefixed value **DLIMT**, the *transport trolley* must be stopped; it will be resumed when *Sonar* detects again a distance higher than **DLIMT**.

The *Led* is used as a *warning devices*, according to the following scheme:

- the *Led* is **off** when the *transport trolley* is at *HOME*
- the *Led* **blinks** while the *transport trolley* is moving
- the *Led* is **on** when *transport trolley* is stopped.

Service users story

The story of the *ColdStorageService* can be summarized as follows:

1. A *Fridge truck* driver uses the *ServiceAcessGUI* to send a request to store its load of **FW** kg. If the request is accepted, the driver drives its truck to the INDOOR of the service, before the ticket exipration time **TICKETTIME**.
2. When the truck is at the INDOOR of the service, the driver uses the *ServiceAcessGUI* to enter the ticket number and waits until the message **charge taken** (sent by the *ColdStorageService*) appears on the *ServiceAcessGUI*. At this point, the truck should leave the INDOOR.
3. When the service accepts a ticket, the *transport trolley* reaches the INDOOR, picks up the food, sends the **charge taken** message and then goes to the *ColdRoom* to store the ffod.
4. When the deposit action is terminated, the *transport trolley* accepts another ticket (if any) or returns to HOME.
5. While the *transport trolley* is moving, the *Alarm requirements* should be satisfied. However, the *transport trolley* should not be stopped if some prefixed amount of time (**MINT** msecs) is not passed from the previous stop.
6. A *Service-manager* migtht use the *ServiceStatusGUI* to see:
   - the **current state** of the *transport trolley* and it **position** in the room;
   - the **current weigth** of the material stored in the *ColdRoom*;
   - the **number of store-requests rejected** since the start of the service.

About requirements

The development of the *ServiceStatusGUI* is optional. However, it is required if the working team is composed of 3 person.