Final **Project**

Autonomous Software Agents
UniTn 2021/2022

Evelyn Turri
https://github.com/EvelynTurri/ProjectASA

Introduction

The house has 2 main floors and a garage. It has a lot of smart devices, most of them work directly through the House Agent, and some of them have their own agent.

The aim is to have a smart house, but looking also for costs, electricity and consumes. In order to save money and to help in the explained aim, on the rooftop there are solar panels and a photovoltaic system.

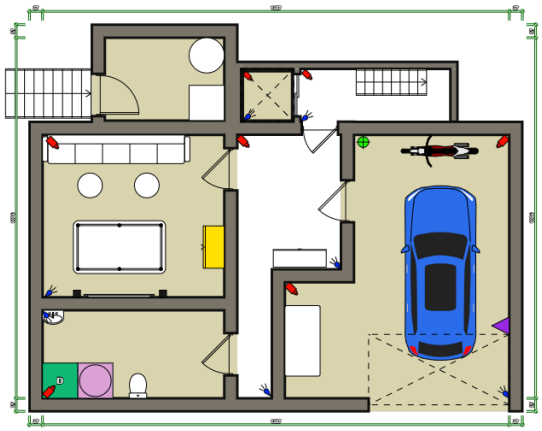
The house is reachable also from people with disability, thanks to the presence of an elevator, and the absence of stairs inside the house and between the rooms.

House description and blueprint

The chosen house has two main floors with an underground floor. There is an elevator and the stairs that connect each floor.

Underground Floor

The underground floor has 3 main rooms: the garage, the laundry and a biliard room. They are all connected thanks to a central hallway, which permits to pass from a room to another without any problem. Clearly, the rooms in the underground floors do not have any windows.



Garage The garage is in the underground floor, and it is used from the residents to park and recharge their electric car. It is connected with a little hallway. It is not heated. It has a main light above the position of the car, and some secondary lights, one above the motorcycle position and other two above the work position on the left wall.

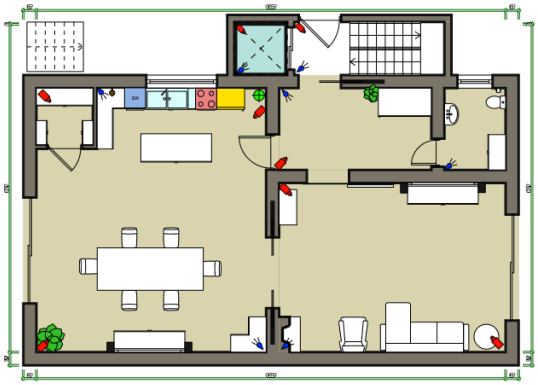
Laundry It is a little room, where the residents have all the possible appliances to do the laundry, as the washing machine and the dryer.

Billiard Room A room with inside a billiard table, a long sofa with 2 round tables and a fridge for the drinks. It has inside also some devices as a TV and two sound-boxes.

Boiler Room The boiler room is accessible only from the outside of the house. There is inside a boiler.

Ground Floor

The ground floor presents an open space with the kitchen and the diving room, a big living room, a bathroom, a pantry positioned near the kitchen and an entryway.



Kitchen & Dining Room The kitchen and the diving room are a big open space. It is pretty bright due to the multiple windows. It is directly connected by a big door with the living room and it is also possible to enter in the pantry. It has also a big french door that gives to outside. For the part of the kitchen there is an induction hob, an oven, a dishwasher, a smart fridge and a coffee maker, while for the part of the diving room there is also a TV and a soundbar. In the kitchen there is also the recharging base for the vacuum cleaner of the ground floor.

Living Room The living room is one of the main part of the ground floor. It is connected with the diving room through a big french door, with the entryway and then it has another french door to go outside in the garden. There is a big TV and two soundbar.

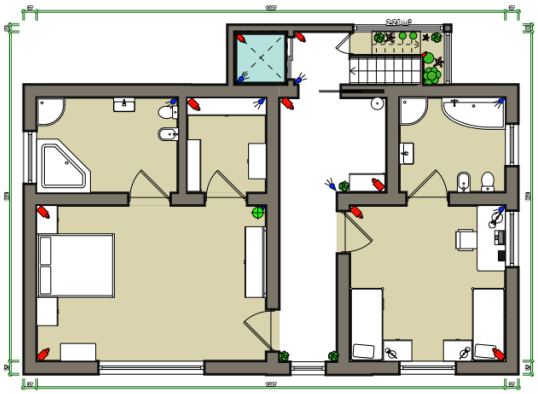
Entryway The entryway is a sort of room that we can find near the main door. It has a closet in order to put the coats when someone just entered in the house.

Ground Bathroom The ground bathroom is connected with the living room by a door. There are inside a toilet and a sink. It has a window and it presents 2 lights, a main light in the middle of the bathroom and the other one attached to the mirror above the sink.

Pantry It is a room addressed to store the supplies. In addition it has a system that keeps track of the stored supplies.

First Floor

The first floor presents the night zone, in particular it has 2 main rooms, a Master Bedroom and a Children Bedroom with 2 bathrooms.



Master Bedroom The Master Bedroom is the main bedroom in the first floor and it is connected with a private bathroom by a door and with the main hallway of the floor. Inside there is a queen-size bed and also a walk-in wardrobe. There is a main light above the bed, 2 more little lights in the side of it, a spotlight near the door and 4 little lights in the wardrobe. There are also some devices, as a TV in front of the bed and a soundbar for the music. The room is really bright, thanks to the big window on the right side of the bed with curtains and roller shutter.

Master Bathroom The Master Bathroom is the private bathroom for the Master Bedroom. They are connected to each other by a door. Inside there are two sinks, a toilet, a shower and a Jacuzzi.

Children Bedroom The children bedroom presents 2 single beds, 2 bedside tables, one desk with its chair and 2 windows. It is connected with the hallway and the children bathroom. It has a main light in the middle of the room, 2 lamps, one for each bedside table, and an additional lamp on the desk.

Children Bathroom The children bathroom is connected only with the children bedroom. Inside there is a rectangle sink, a toilet, a bidet, a shower and a corner bathtube. There are different lights, a main light in the middle of the bathroom, two little lights above the sink and one over the bathtube.

Greenhouse The greenhouse is a corner room, located near the stairs. It is very bright thanks to the windows as big as the walls where they are located. Inside there different types of plants. There are some important devices, different from the other devices in the all the house. There is a thermostat as in the rest of the house, and in addition a humidity controller. In order to keep the ground of the soil of the plants enough wet, there is a controller for it and a smart drip irrigation system in each plant.

Devices

Internal general devices

Door Lockers It is on the door of the main entrance in the ground floor, on the doors near the elevator and the stairs in each floor and on the garage. Its statuses are: `activated` and `deactivated`. The actions are:

- `lock` : It locks the door
- `unlock` : It unlocks the door

Lights There is a light in each room of the house. Their status can be `on`, `off` and they can change color.
The actions for the device are:

- `switchOnLight` : Switch on the light
- `switchOffLight` : Switch off the light

- `setColor` : Change color between normal light, warm white, cold white, red, blue, light pink.
- `eveningMode` : whihch set the color of the light to warm white.

Each light consumes 20W of electricity when switched on.

Roller Shutter The roller shutter are on each window of the house. Their status can be `up`, `down`.
The actions for the device are:

- `liftUp` : Lift the shutters up.
- `pullDown` : Close the shutters.

Camera **Red Device** - Inside every room (a part from the bathrooms, for privacy reasons) there is a camera. The camera can have these following status: `on` and `off`. The related actions are:

- `switch_on` : It switches on
- `switch_off` : It switches off

Moving Sensor **Blue Device** - There is a moving sensor in each room. It can be `activated` or `deactivated`. The actions that can take are:

- `activate` : The sensor activates.
- `deactivate` : The sensor deactivates.
- `alert` : It perceives if someone is moving and it sets the property `perceive` to `true`.

Alarm System All the house is provided of an alarm system, which keeps the house safe. It can be `active` or `inactive`.
The actions are:

- `activate` : The system activates.
- `inactivate` : The system inactivates.
- `alert` : It sound when it is necessary.

Vacuum Cleaner **Light Green Device** - Each floor has its own vacuum cleaner, that can be: `on`, `off`, `charging`, `full_charged` (if the battery is full charged), `charged` (if the battery is up to 50%), `out_of_battery` (if the battery is lower than the 15%) or `cleaning` (if it is cleaning the floor).
It can do also some actions, as:

- `switchOn` : Switch on.
- `switchOff` : Switch off.
- `clean` : Clean if it is dirty.
- `move` : Moves from a room to another.
- `leaveChargerBase` : It leaves the charger base.
- `returnChargerBase` : It returns to the charger base.

Charging Base Vacuum Cleaner **Light Green Device** - The charging base of the vacuum cleaner is in each floor, and it has the capability of recharge the vacuum cleaner. Its statuses can be: `on`, `off` and `recharging`. Its actions are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `sendVacuum` : It sends the vacuum cleaner out of the base

- `receiveVacuum` : It receive the vacuum cleaner in the base

Thermostat There is a thermostat in each room. The actions that can take are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `readTemperature` : It reads the temperature of the room
- `setTemperature` : It sets the desired temperature

External general devices

Photovoltaic System On the rooftop there is a photovoltaic system, a classic one of 3kWp. Its role is the one to provide electricity at the house. The system produces 4000kWh/year.

Solar Panels There are 2 solar panels, each of $3m^2$. They provide the hot water for the sanitary part of the house. And they heat up an accumulation of water of the boiler room. the solar panels can provide an average of 1,5kW in the sunny days, while during the night 0kW.

Underground Floor

Electric Car The car in the garage has a capacity of 64KWH, and can be charged at 3.7kW.
Status of the car includes

- `car_in_garage` : The car is in the garage
- `car_not_in_garage` : The car is not in the garage
- `fully_charged`
- `half_charged`
- `need_recharged`

The actions are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `startCharge` : It starts charging

Recharge Column Base **Purple Device** - It is the recharging column for the electric car. It can recharge the car at 3.7kW maximum, and if needed can recharge the car at a lower power.

Washing Machine **Pink Device** - The washing machine status can be: `on` and `off`. As actions have:

- `switchOn` : it switches on
- `switchOff` : it switches off
- `start` : it starts the selected program

Dryer **Green Device** - As the washing machine the status of the dryer can be: `on` and `off`. As actions have:

- `switchOn` : it switches on
- `switchOff` : it switches off
- `start` : it starts the selected program

Ground Floor

Induction Hob **Light Red Device** - The induction hob can be `on` and `off`. Moreover because it has 4 places (single hob), each of them has also the status `heating_#` (The number # is working and it has over a pan). And when the general status is `full_heating` it means that all the places are working.

Single Hob It is a single hob of the induction hob. Each of them has the status `on`, `off` and a value with the level of heating.

Oven **Under the Hob** - The oven has the status: `on`, `off`, a vale for the `temperature` and `full` (if there are inside elements). The actions are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `setTemperature` : It sets the temperature at a certain value

Smart Fridge **Yellow Device** - The smart fridge is a smart device, it can have different statuses as `on`, `off`, a value for the `temperature`. The smart fridge has its own memory, which keeps track of all the elements added in the fridge. The possible actions are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `setTemperature` : It sets the temperature at a the desired value
- `addElement` : It adds a certain element in its memory
- `removeElement` : It removes a certain elements from its memory
- `cleanMemory` : It resets all the memory

Dishwasher **Indigo Device** - The dishwasher has the statuses `on`, `off`, `running`, `finished`, `full` and `empty`. The possible actions are:

- `switchOn` : It switches on
- `switchOff` : It switches on
- `start` : It starts cleaning

Coffee Maker **Orange Device** - The coffee maker is a little tool located in the kitchen. The function is only one and it is to prepare coffee. Its statuses are: `on`, `off`, `making_coffee` and `finished`. The possible actions are:

- `switchOn` : It switches on
- `switchOff` : It switches on
- `start` : It starts to make coffee
- `finish` : It alerts when the coffee is ready

First Floor

Humidity Controller It checks the humidity in the greenhouse. Its statuses can be: `on` and `off`. The possible actions are:

- `switchOn` : It switches on
- `switchOff` : It switches off
- `setHumidity` : It sets the humidity to a certain value

Metrics

Hot Water

Thanks to the solar panels, the family has an advantage of pretty much the 75% in a year.

Electricity

Thank to the photovoltaic system, the resident are able to save pretty much 300€/year. This because they if they directly consumes the electricity, they directly use pretty much 1200kW, which is the 30% of 4000. Due to the fact that the electricity costs 0.25€/kWh, and they save 1200kW, it means that they save 0.25€/kWh x 1200kW = 300€/year. Directly use the electricity means that they should make work the dryer, the washing machine and so on, during the day when the photovoltaic system is working, and not during the night.

Cleaning Time

The vacuum cleaner takes different time for cleaning different rooms.

In the underground floor the robot takes 1.5h to clean the garage, 0.5h to clean the billiard room and 0.25 to clean the bathroom, so in total it takes 2 hours to clean all the underground floor.

In the ground floor it takes 0.5h to clean the living room, 1 hour to clean the kitchen and diving room, while only 0.4h to clean the entryway and the bathroom. SO for cleaning all the ground floor it takes pretty much 2 hours.

In the first floor the vacuum cleaner takes 0.75h to clean the Master Bedroom, 0.25h for the Master Bathroom, 0.5 for the Children Bedroom and 0.2 for the Children Bathroom. In total it takes pretty much 2.5h for the total floor.

People and Agents

This section presents intelligent and autonomous entities in the house, including people and agents.

People

Residents in the house include a couple, Mark and Rose, with two children, Peter and Lili.
From Monday to Friday, Mark goes to work by motorbike, he leaves the house at 7 a.m. and he comes back at 5 p.m., while Rose leaves the house with the 2 children at 8 a.m., and she brings the children to school by car.
Roses comes back for lunch at 2 p.m., and then she leaves again at 4 p.m., because she has to go to take the children and bring them to do sport for one hour, so that they are back at home by 6 p.m..
The children go to sleep by 9 p.m. while Mark and Rose by 11.30 p.m.. After 9 p.m., they usually stay in the living room watching a movie, until they go to bed.
On Saturday they usually stay at home, while on Sunday they usually go on a one-day trip by car. this means that the resident need the car full charged for Sunday, and that the car can charge on Saturday.

Agents

The agents are divided in **Non-Planning Agents** (also said ”Stupid agents” for *Assignment 2*) and **Planning Agents** (for *Assignment 3*).

Non-Planning Agents

House Agent

The agent assists residents by taking autonomous decisions, while still being responsive to residents' behaviors. It tries to minimize energy consumption/cost.

Lights The house agent controls the lights of the house. The lights are connected with the moving sensor. If the moving sensor perceives that someone is moving in a room, the light of that room will switch on, if it is not switched on yet. In this part it is explain how the house agent controls the lights of the house.

Roller Shutters The house agent will automatically lift up the shutter at 6 a.m., when the resident will get up, and as well it will pull down at 8 p.m..

Devices' Agents

Coffee Maker Agent It is able to switch on the coffee maker at a certain time, in particular from Monday to Friday the agent will switch on the device and actually make coffee at 6 a.m., when the residents get up.

Planning Agents

In the house there are different planning agents, some of them also communicates between each other. There are:

- Three vacuum cleaners, one for each floor
- A charging base for each vacuum cleaner
- A door locker agent
- A security system agent

Each vacuum cleaner communicates with its charging base, while the door locker agents communicates with the security system agent.

Vacuum Cleaner Agents This agent controls the vacuum cleaner and it communicates with the charging base agent in order to have more efficiency in the all process. First of all the vacuum cleaner has to clean the designed floor every day, in order to have the house as clean as possible in every moment.
As said in the introduction of the planning agents, there are three vacuum cleaners agents, named:

```
vaccumCleanerUnderground : new VaccumCleaner(this, 'vaccum_cleaner_underground', this.floors.undergroundFloor),
vaccumCleanerGroundFloor : new VaccumCleaner(this, 'vaccum_cleaner_groundfloor', this.floors.groundFloor),
vaccumCleanerFirstFloor : new VaccumCleaner(this, 'vaccum_cleaner_firstfloor', this.floors.firstFloor),
```

The goal for each of them is in the following form:

- `goal: ['clean_underground_floor vaccum_cleaner_underground']`
- `goal: ['clean_ground_floor vaccum_cleaner_groundfloor']`
- `goal: ['clean_first_floor vaccum_cleaner_firstfloor']`

With only one line of code and with this goal, the PDDL of the vacuum cleaner is able to:

- Switch on the vacuum cleaner
- Leave the charger base

- Move in each room
- If the room is dirty, clean it before to leave it
- When it has finished, send a message to the charger base
- Then the charger base agent gives the command to the vacuum cleaner to return back to the base
- When the vacuum cleaner gets to the base, it switches off

The agent knows if the vacuum cleaner is switched on/off, if it is in the base or not, and it knows the different connections between the rooms.

Charging Base Agents The charging base agent does its actions when the vacuum cleaner agent gives the command to do them. Specifically when the vacuum cleaner knows that all the rooms of a floor are cleaned, it sends a message to the charger base agent, in order to update the beliefs of the charger base and makes the vacuum cleaner comes back to the base, indeed the charging base agent sends back a message to the vacuum cleaner agent with a new goal, to make it go back to the charger base and switch off it.

Doors Locker Agent The door agent locks the doors provided of the automatic lockers. When the resident leaves the house, meaning that there are no person in the house the door agent will lock all the doors. The door will be locked also when all the family go to sleep, so at 11 p.m..
Because the doors locker has to send a message to the security system agent to activate or deactivate the alarm, the domain presents 4 actions for this type of planning: `LockDoors`, `UnlockDoors`, `LockCheckStatus` and `UnlockCheckStatus`.
Usually the final goals are to activate and deactivate the doors locker, but in order to achieve them, the agent has to work also on the security system, indeed if it has to lock the doors, it has to activate the alarm before to achieve the goal, so the action `LockCheckStatus` is an intermediate action, in order to be able to activate the security system, before to apply the effect of the action `LockDoors`. For the goal of unlock the doors, it works the same, with the only difference that before of unlocking the doors, the security system has to deactivate the alarm.

Security System Agent The security system agent activates the alarm system when the doors are locked. So the agent will work only when it receives a new goal from the doors agent, in order to be sure that the security system will be activated before locking the doors and will be deactivated before unlocking the doors.
The security system agent knows if the alarm is already activated or not.

Implementation

Sensors and Agents

Sensors and Agents perception

The non-planning agents perceive through the intentions and the `notifyChange` function, that notifies every time there is a change of a certain status.
While the planning agents build their perceptions based on the beliefs states. The main beliefs state are initialized at the beginning, but during the planning every agent updates its beliefs, in order to be able to take another action.

Agents acting in a shared environment

Most of the agents take autonomous decisions, but some of them take actions only by waiting a message from the others, for example as the charging base of the vacuum cleaner or the security system.
As described in the previous section, each planning agent has its own actions, which are implemented in order to build a plan.

Agent interaction and coordination

Most of the agents communicate between each other, in order to reach better the goal and a most complicated one.
The communication works through the MessageDispatcher and the Postman. Postman intention and subgoal has to push for both the sender and the receiver.
The agents that communicate between each other are:

- The vacuum cleaner agent and the charging base agent
- The doors locker agent and the security system agent

Other two intention, that work together and they communicate in a sort of sense are the lights intention and the moving sensors intention, even if the agent is for both the house agent.

Scenario

In the code there are different scenario, in each of them different agents are working in the house.

For more details about the `log` and the different scenarios, look at the files in the source code, in the subfolder `./logs`:

- `Scenario1.log`
- `Scenario2.log`
- `Scenario3.log`

It is possible to run also another possible scenario, called `myworld.js`, where there are the movement of the person, and the different agents all together.

Scenario 1 In the Scenario 1 there is a typical day for Mark and Rose, with the implementation of some ”stupid” agents. In this scenario there are only 2 agents, which take different actions:

1. The house agent
2. The coffee maker agent

The house agent controls:

- The lights of the house
- The moving sensors of the house
- The roller shutters

While the coffee maker agent controls the coffee maker device.

We start from the house agents and its actions.
The bigger part of work is for the lights and the moving sensors, which in a certain sense communicate between each other, indeed the house agent is able to understand when a person enters in a room because the moving sensors of each room are triggered. When the moving sensors is triggered, the house agent understands its change of status and through the lights intentions, is able to switch on the light of a certain room, whenever a moving sensor is activated for the first time in a certain room. Below I report some logs for the actions of moving a person in a certain room, triggered the moving sensor and switch on the light.

```
Mark moves from the master bedroom to the kitchen
Mark      moved from master_bedroom      to  hallway_firstfloor
Mark      moved from hallway_firstfloor  to  landing_firstfloor
Mark      moved from landing_firstfloor  to  landing_ground
```

```
Mark      moved from landing_ground      to  entryway
Mark      moved from entryway  to  kitchen
house_agent>     MovingSensorsIntention#0
                 sense:  sensor moving_sensor_hallway_firstfloor perceives true
house_agent>     MovingSensorsIntention#0
                 sense:  sensor moving_sensor_landing_firstfloor perceives true
house_agent>     MovingSensorsIntention#0
                 sense:  sensor moving_sensor_landing_ground perceives true
house_agent>     MovingSensorsIntention#0
                 sense:  sensor moving_sensor_entryway perceives true
house_agent>     MovingSensorsIntention#0
                 sense:  sensor moving_sensor_kitchen perceives true
house_agent>     LightIntention#8    sense: light light_kitchen switched on
```

The light that switches on, is the one of the room when Mark arrives, due to the fact that the last notify change overwrite the previous one, so we can see in the log only the last one.
The movement of Mark is simulated manually with the function `move` of the class `Person`.

For the part of the roller shutter instead, they lift up at 6 a.m. and they pull down at 8 p.m.. The simulation is not done manually, but only with the intention for the roller shutters, because it makes a control on the hour of the global clock. Two lines for the log for the lift up action and the pull down action are reported below:

```
house_agent>     AllRollerShutterIntention#1 sense:It's 6 a.m.! Roller shutter are up
house_agent>     AllRollerShutterIntention#1 sense:It's 9 p.m.! Roller shutter are down
```

While the second agent is for the coffee maker. It just switches on the coffee maker device at 6 a.m. and it finishes at 6:15 a.m. to prepare coffee and then it switches off.

Some lines of log for the agent are:

```
coffee_maker_agent> CoffeeMakerIntention#22 sense: It's 6 a.m.! Coffee Maker switches
                                            on and starts to make coffee

coffee_maker_agent> CoffeeMakerIntention#22 sense: The coffee is ready and the coffee
                                            maker switches off
```

Scenario 2 In the Scenario 2, there are all the vacuum cleaners working in the house. The one in the first floor is starting at 8 a.m., when all the family is out of home, while the others two are working during the night.
In this kind of scenario there are 6 different agents:

1. `vacuum_cleaner_underground`
2. `vacuum_cleaner_groundfloor`
3. `vacuum_cleaner_firstfloor`
4. `charging_base_underground`
5. `charging_base_groundfloor`
6. `charging_base_firstfloor`

These agents work in couple: the vacuum and the charger base undergorund, the vacuum and the charger base in the ground floor and the two in the first floor.

Below we take as example only one of the couple (`vacuum_cleaner_underground` and `charging_base_underground`), due to the fact that they work in the same way, a part from some specific functions, which will be explained better at the end of the paragraph.

For each of the vacuum cleaner there are some initial beliefs, some of them to build the basic knowledge and some others to tell to the agent in which state is the vacuum cleaner.
Below there are the main initial beliefs:

```
house.devices.vaccumCleanerUnderground.beliefs.declare('in_room vaccum_cleaner_underground garage')
house.devices.vaccumCleanerUnderground.beliefs.declare('switched_off vaccum_cleaner_underground')
house.devices.vaccumCleanerUnderground.beliefs.declare('in_base vaccum_cleaner_underground')
```

Then at the desired hour there is the line of code with the subgoal:

```
house.devices.vaccumCleanerUnderground.postSubGoal(new RetryGoal_vc( { goal: new PlanningGoal( { goal: ['clean_underground_floor vaccum_cleaner_underground'] } ) } ))
```

For each vacuum cleaner 4 plans are built, two from the vacuum cleaner agent and the other two from the charging base agent.
First of all the vacuum cleaner agent builds the first and most complicated plan; the log is reported below:

```
Plan found:
- (switchon vaccum_cleaner_underground)
- (leavechargerbase vaccum_cleaner_underground)
- (clean vaccum_cleaner_underground garage)
- (move vaccum_cleaner_underground garage hallway_underground)
- (clean vaccum_cleaner_underground hallway_underground)
- (move vaccum_cleaner_underground hallway_underground laudry)
- (clean vaccum_cleaner_underground laudry)
- (move vaccum_cleaner_underground laudry hallway_underground)
- (move vaccum_cleaner_underground hallway_underground billiard_room)
- (clean vaccum_cleaner_underground billiard_room)
- (move vaccum_cleaner_underground billiard_room hallway_underground)
- (move vaccum_cleaner_underground hallway_underground landing_underground)
- (clean vaccum_cleaner_underground landing_underground)
- (cleanallroomsunderground vaccum_cleaner_underground garage laudry
    billiard_room hallway_underground landing_underground)
- (cleanundergroundfloor vaccum_cleaner_underground)
```

As we can see in the plan, the vacuum cleaner clean the room where it is, if it is dirty, otherwise it moves in the next room. If during the journey from a room to another, it crosses different rooms already cleaned, obviously it does clean them again.
When the vacuum cleaner leaves the charger base, it automatically communicates with the connected charger base, in order to update the beliefs also of the charger base. The message is sent throw this action:

```
class LeaveChargerBase extends Check {
    static parameters = ['obj'];
    static precondition = [ ['vaccum', 'obj'], ['switched_on', 'obj'], ['in_base', 'obj']];
    static effect = [ ['not_in_base', 'obj'] ];
    *exec ({obj}=parameters) {
        yield this.checkPreconditionAndApplyEffect()
            this.agent.leaveChargerBase();
            this.agent.beliefs.declare('not_in_base' + ' ' + obj);
            this.agent.beliefs.undeclare('in_base' + ' ' + obj)
            if (this.agent.name == 'vaccum_cleaner_underground') {
                MessageDispatcher.authenticate(this.agent).sendTo('charging_base_underground', new PlanningGoal( { goal: ['out vaccum_cleaner_underground charging_base_underground'] } ))
            }
            if (this.agent.name == 'vaccum_cleaner_groundfloor') {
                MessageDispatcher.authenticate(this.agent).sendTo('charging_base_groundfloor', new PlanningGoal( { goal : ['out vaccum_cleaner_groundfloor charging_base_groundfloor'] } ))
            }
            if (this.agent.name == 'vaccum_cleaner_firstfloor') {
                MessageDispatcher.authenticate(this.agent).sendTo('charging_base_firstfloor', new PlanningGoal( { goal: ['out vaccum_cleaner_firstfloor charging_base_firstfloor'] } ))
            }


    }
}
```

And the other agent will receive the message, and the resulting log is:

```
charging_base_underground > PostmanAcceptAllRequest#3 Reading received message
charging_base_underground   Trying to use intention OnlinePlanning to
                            achieve goal PddlGoal#70[object Object]
charging_base_underground > OnlinePlanning#70 Intention started
```

The log of the resulting plan is following for the agent `charging_base_underground`:

```
Plan found:
- (outchargerbase charging_base_underground vaccum_cleaner_underground)
```

Then, when the vacuum cleaner finishes to clean all the rooms in the floor, it sends another message to the charging base agent in order to make the vacuum cleaner comes back to the base and starts a new goal (similar to the previous one), with the only difference that the effect is to have the vacuum cleaner in the base. Just few lines of code are needed to make it works:

```
class inChargerBase extends Check {
    static parameters = ['obj', 'v'];
    static precondition = [ ['charger_base', 'obj'], ['vaccum', 'v'], ['out', 'v', 'obj']];
    static effect = [ ['in', 'v', 'obj']];
    *exec ({obj, v}=parameters) {
        yield this.checkPreconditionAndApplyEffect()
            this.agent.receiveVaccum();
            this.agent.beliefs.declare('in' + ' ' + v + ' ' + obj), this.agent.beliefs.undeclare('out' + ' ' + v + ' ' + obj);
            MessageDispatcher.authenticate(this.agent).sendTo(v, new PlanningGoal({goal: [`switched_off ${v}`]}))
    }
}
```

As there is in the previous code, the charging base agent sends another message to the vacuum cleaner agent, in order to start another plan and make the vacuum cleaner comes back to the base and switches off.

The `vacuum_cleaner_underground` agent will receive the message and will try to start a new plan, indeed the log it will be in this form:

```
vaccum_cleaner_underground > PostmanAcceptAllRequest#0 Reading received message
vaccum_cleaner_underground   Trying to use intention OnlinePlanning to achieve
                             goal PddlGoal#77[object Object]
vaccum_cleaner_underground > OnlinePlanning#77 Intention started
```

With the new plan in this form:

```
Plan found:
- (returnchargerbase vaccum_cleaner_underground)
- (switchoff vaccum_cleaner_underground)
```

In this way with only one line of code, as shown in the postGoal code, we are able to make the vacuum cleaner cleans all the floor, switches on and also switches off.

Note : as said at the beginning of the paragraph, some functions differ for each vacuum cleaner. In the planning file of the vacuum cleaner agent, there are three different actions for cleaning all the different floors and for the final goal, where it checks if the rooms are all cleaned. This is due to the fact that the floors have different rooms, different connections, and especially different number of rooms.

Scenario 3 In the Scenario 2 there is the simulation of the doors locker agent and the security system agent. They are working together, and the second waits fro commands from the first one.
As simulation the doors lock at 8 a.m., when Rose with the children leave the house and no one is in the house. While they

will unlock at 2 p.m. when Rose comes back home.

In this scenario there only 2 agents:

1. `doors_locker`
2. `alarm_system`

There are two types of goal, one for unlocking the doors, and the other one for locking them. In the following lines of codes it is shown how to request the two goals to the `doors_locker` agent.

```
house.devices.doorsLocker.postSubGoal(new RetryGoal_vc( { goal: new PlanningGoal( { goal: ['activate house_door_lockers'] } ) } ))
house.devices.doorsLocker.postSubGoal(new RetryGoal_vc( { goal: new PlanningGoal( { goal: ['deactivate house_door_lockers'] } ) } ))
```

While the only main initial beliefs for both agents are very few:

```
house.devices.doorsLocker.beliefs.declare('deactivate house_door_lockers')
house.devices.allarmSystem.beliefs.declare('deactivate house_security_system')
```

For each of the 2 goals, 2 plans are built, one for the doors locker agent, and the other one for the security system agent. It follows the complete log for the doors lock goal. The following part is for the `doors_locker` agent.

```
doors_locker    Trying to use intention RetryFourTimesIntention_vc to achieve
                goal RetryGoal_vc#3
doors_locker>   RetryFourTimesIntention_vc#2    Intention started
doors_locker    Trying to use intention OnlinePlanning to achieve goal PddlGoal#2
doors_locker>   OnlinePlanning#3   Intention started
doors_locker>   OnlinePlanning#3   Plan found:
doors_locker>   OnlinePlanning#3   - (lockcheckstatus house_door_lockers
                                       house_security_system)
doors_locker>   OnlinePlanning#3   - (lockdoors house_door_lockers)
doors_locker>   OnlinePlanning#3   Starting sequential step
                (LockCheckStatus house_door_lockers house_security_system)
                Effect: lock_check_status house_door_lockers
doors_locker>   LockCheckStatus#4   Intention started
doors_locker    Belief changed: lock_check_status house_door_lockers
doors_locker    Belief changed: not unlock_check_status house_door_lockers
```

While the following log are for the `alarm_system` agent:

```
allarm_system>  PostmanAcceptAllRequest#0   Reading received message PddlGoal#6
allarm_system   Trying to use intention OnlinePlanning to achieve goal PddlGoal#6
allarm_system>  OnlinePlanning#6    Intention started
allarm_system>  OnlinePlanning#6    Plan found:
allarm_system>  OnlinePlanning#6    - (activate house_security_system)
allarm_system>  OnlinePlanning#6    Starting sequential step
                                    (Activate house_security_system)
                                    Effect: activate house_security_system
allarm_system>  Activate#7          Intention started
allarm_system   Belief changed: activate house_security_system
allarm_system   Belief changed: not deactivate house_security_system
```

```
allarm_system>  Activate#7          Intention success
allarm_system>  OnlinePlanning#6    Intention success
allarm_system   Succesfully used intention OnlinePlanning to achieve goal PddlGoal#6
```

Source code organization

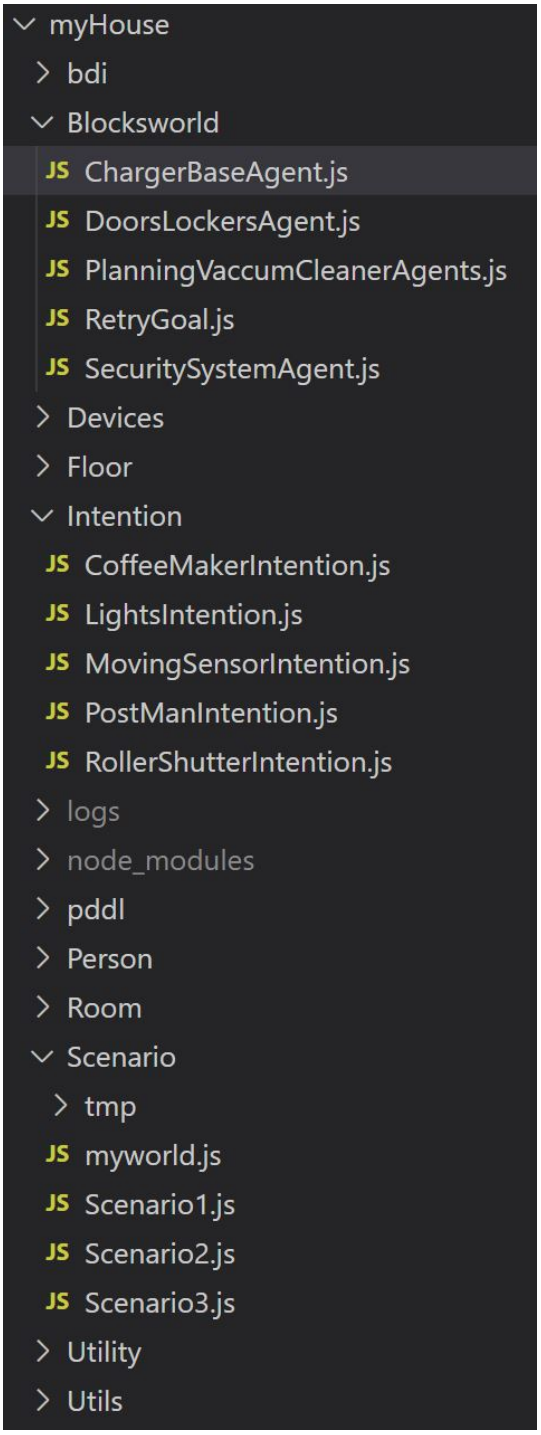
The folder of the source code organization is divided in 12 subfolders (`node_modules` subfolder is not counted).

The files in the subfolders `Blocksworld`, `Intention` and `Scenario` are highlighted, becouse the first subfolder contains all the file for the planning agents, while the second one contains the inetion of the other agents and in the last one there are the different 3 types of scenario, and one `myworld.js` file, where there are all the scenario put together.

In the subfolders: `Floor`, `Person` and `Room` there are all the classes needed to build the House (file in the `Room` subfolder).

While in the `Device` subfolder there are all the devices divided again based on the floor or if they are general devices.

It is important to say that the planning agents are directly connected with the corresponding devices, thanks to the fact that each device does not extends the `Observable` class, but they extends directly the `Agent` class.
This helps in updating the beliefs and pushing the intention or the goals, because we are working directly on the device.



NOTE : The github link of the project is provided at the beginning of the report.