



## **Intelligens elosztott rendszerek**

BMEVIMIAC02

2022

Varga Tamás (BFLOHU)

Gyulai Gergő László (JJY2VO)

Dömötör Péter Balázs (G2Y5TI)

# Okos otthon: takarító porszívók (Cleaner)

# Tartalomjegyzék

Tartalomjegyzék .....	2
A feladat leírása.....	3
A megoldás összefoglalása.....	4
Ágensek: .....	4
Szenzor:.....	4
Porszívók:.....	4
Kontroller:.....	4
Rendszer összefoglaló ábrája: .....	5
Fejlesztés összefoglalása: .....	6
Használt Jason verzió:.....	6
ASL szinten készült: .....	6
Java szinten készült: .....	6
Készített program ismertetése .....	7
Szimuláció.....	7
Cleaner.asl.....	7
Controller.asl .....	8
Sensor.asl .....	8
Demo videó linkje:.....	8

# A feladat leírása

Feladatunkban egy okos ház takarító rendszerét valósítottuk meg. A szobát egy szenzor végig pásztázza, ha talál szennyeződést akkor azt elküldi a kontrollernek. A kontroller az okos porszívók között levezényel egy aukciót, az ágensek egyénileg a szennyeződés és a saját Manhattan távolságuk alapján licitálnak. Amikor egy okos porszívó tele lesz szennyeződéssel akkor elvonul a dokkolójára kiüríteni magát.

A helyszín egy nagy méretű okos otthon, amelyet a robotok által értelmezhető négyzethálóra osztottunk fel.

# A megoldás összefoglalása

## Ágensek:

### Szenzor:

Folyamatosan szkenneli a szobát amennyiben koszt talál elküldi a kosz koordinátáit és mennyiségét a kontrollernek.

### Porszívók:

Önműködő robot porszívók a kontrollerrel kommunikálva megkapják a kosz helyét és odamennek felszedni. A kosz felszedéséért licitálnak, amit a kontroller vezényel le. A porszívóból 6 darab van 2 nagyobb és 4 kisebb kapacitással rendelkeznek. Amikor betelnek elmennek a központi dokkolóhoz és kiürítik a felgyűjtött szennyeződést.

### Kontroller:

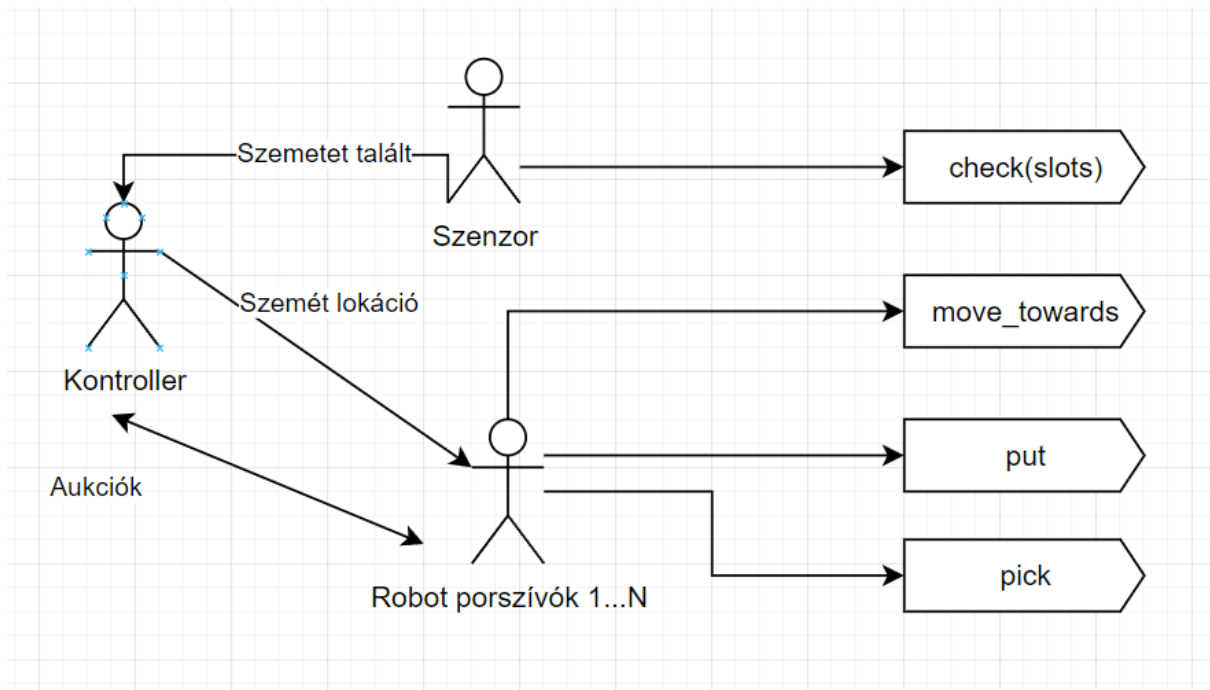
Egy példány létezik és a többi ágens ezzel kommunikál. A szemét hollétéről a szkennertől kapott információkat továbbítja az összes robot porszívónak. Licitálással eldönti a robot porszívók között, hogy melyikük melyik szemetet takarítsa fel.

A robot porszívók a takarítás feladatának kiosztásáért egymással versengenek.

A licitálásuk alapja a saját és szennyeződés pozíciója közötti Manhattan távolság.

A szennyeződés a térben idővel növekszik.

## Rendszer összefoglaló ábrája:



# Fejlesztés összefoglalása:

A fejlesztés elején megoldottuk hogy a projektet gradle projektént lehessen futtatni. A kirajzoláshoz a tárgy oldalon javasolt RescueFramework keretrendszerből emeltünk át elemeket, mivel így esztétikusabb külsőt tudtunk adni a programunknak, és az átállásnak köszönhetően megszűnt a zavaró villódzás.

Használt Jason verzió:

3.0

A futtatáshoz Java 16+ verzióra van szükség.

A projekt futtatható Gradle-ből is.

ASL szinten készült:

Az ágensek belső működése, valamint a közöttük zajló kommunikáció teljes egészében AgentSpeak környezetben valósult meg.

Java szinten készült:

A java szinten írt osztályok felelősek a felhasználói felület megjelenítéséért, aminek segítségével megfigyelhetjük a rendszer működését. Illetve a szoba és a benne lévő szennyeződések kezeléseket is a Java oldal kezeli.

# Készített program ismertetése

## Szimuláció

A HouseEnv Java osztály felelős a szükséges extra Asl-ből hívható parancsok megvalósításáért.

Extra parancsok:

- Next: A szenzor használja a következő cellára való lépéshez.
- Put : A robot tartályát üríti ki.
- Pick: A robot használja szennyeződés felvételére.
- Move\_towards: A robot használja, a cél cella eléréséhez

Továbbá a HouseEnv felelős az ágensek percepcióinak a kezeléséért:

- Pozíció minden ágensnek
- Szenzornak a cella tartalma
- Robot tartály telítettsége.

A PaintPanel segéd osztállyal rajzoljuk ki a képernyőre a szimulációt. Ez egy JPanel-ből származtatott osztály ami a terület rácsát a benne elhelyezkedő ágensekkel együtt jeleníti meg. A szenzort türkizkékkel a robotokat zölddel jeleníti meg.

A szoba állapotát a HouseWorldModel osztály tárolja. A szennyeződés állapotát egy két dimenziós tömbben tároljuk amiben az értékek jelenleg 0-4 lehetségesek. A nulla érték kosz mentes területet jelöl.

A Manhattan távolság kiszámolására létrehoztunk egy ManhattanDistance osztály-t ami a DefaultInternalAction-ből származik és egy custom action-t implementál.

## Cleaner.asl

Jelenleg két fajta robot porszívónk létezik: 4db 5-ös tartály méretű és 2db 15-ös tartály méretű, de mennyiségük és tartály méretük szabadon konfigurálható. A robot porszívó tartály mérete meghatározza, hogy mennyi szennyeződés fér bele. Ha egy Robot porszívónak nincs több tárhelye, akkor visszatér a kezdő pozícióba ahol kiüríti a szennyeződést. A robot képes szabadon mozogni a szimulációs térben, valamint a Szenzor által detektált szemetekre tud licitálni a többi robot porszívóval versenyezve. A licitálás értékét a közte és a szennyeződés közt lévő Manhattan távolságból számolja ki.

## Controller.asl

A rendszerben egyetlen controller létezik. Ő felelős a szenzor és a robotok közötti kommunikációért, valamint a licitálások lebonyolításáért. Ha a szenzortól megkapja a szemét helyéről az információt, továbbítja azt az összes robot felé. A robotok ezt követően licitálni kezdenek a szemétre, amelyből a controller állapítja meg a győztest, akit ezt követően értesít.

## Sensor.asl

Egy szenzor létezik, aminek feladata a megadott terület folyamatos bejárása, valamint a szemetek detektálása. Ha szemetet talál értesíti róla a kontrollert, majd tovább folytatja a keresést.

## Demo videó linkje:

<https://youtu.be/cHPP90yHph8>