

# Raktárrendszer

IER Házi feladat

2021/22/tavaszi félév

Csapattagok:

Gál Gyula  
AQBQED

Lipták- Lukácsik Kolos  
D68PQ6

Varga Balázs  
E90HGZ



M Ű E G Y E T E M 1 7 8 2

# Tartalomjegyzék

<b>Feladat leírása</b>	<b>3</b>
<b>A megoldás összefoglalása</b>	<b>4</b>
1. Ágensek	4
2. Ágensek kapcsolatai	4
3. Ágensek viselkedése	5
<b>A fejlesztés összefoglalása</b>	<b>5</b>
1. Milyen Jason verziót használtak?	5
2. Mi készült Java szinten?	5
<b>A kifejlesztett program ismertetése</b>	<b>7</b>
1. Program kinézete	7
2. Kezelőfelület	7
3. Ágens működése	9
3.1. Fájlok funkcionálitása	9
4. Java oldal működése	9
4.1. Fájlok funkcionálitása	9
4.2. Projektfájl	10
5. Futásidőben módosítható paraméterek	10
6. Ágens programok rövid összefoglalása (BDI jelleggel)	11
7. A program egészének összefoglalása (többágenses rendszer jelleggel)	11
8. Link a videóhoz	11

## Feladat leírása

A feladat témája egy raktár kezelő rendszer működése. A rendszerbe érkeznek különböző csomagok, melyeket egy, vagy állíthatóan több szelektáló ágens típus szerint szétválogat külső beavatkozás nélkül. Amint egy csomag megérkezik, a szelektáló ágens/ágensek elkezdik elemzését és az adott ágens állapotának megfelelően a helyére viszi a bejövő csomagot. Ha egy csomagot a helyére vittek az ágensek, akkor az eltűnik.

A csomagok érkezésének sebessége a GUI-n keresztül növelhető, illetve csökkenthető lesz. Ez felveti a lehetőségét, hogy az ágensek nem bírják el a beérkező csomagok mennyiségével. Ilyenkor egy adott mennyiségű nem helyre rakott csomag után nem érkezik újabb, amíg egyet legalább a helyére nem raktak az ágensek.

# A megoldás összefoglalása

## 1. Ágensek

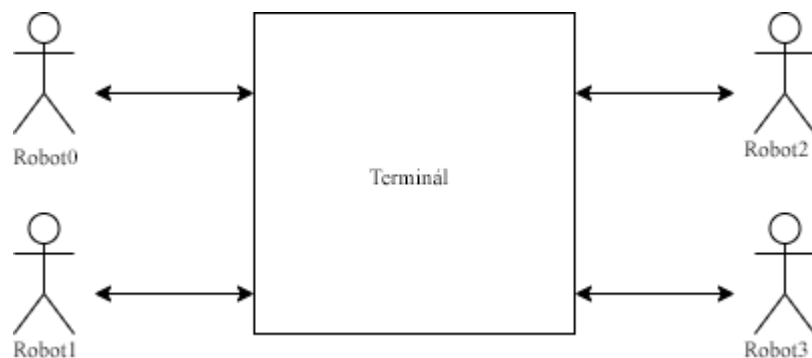
- **Terminál:**

A terminál tudja magáról, hogy mi hol helyezkedik el benne. Ezt az információt a Robotok számára teszi elérhetővé, azaz tudatja velük, hogy a legrövidebb utat, a hozzájuk legközelebbi csomaghoz, illetve piros és kék lerakási ponthoz.

- **Robot:**

Robotok. Feladatuk a futószalag mellett megjelenő különböző színű dobozokat a különböző színű raktárakba vitele.

## 2. Ágensek kapcsolatai



- **Terminál:**

A terminál mindennel kapcsolatban áll. A benne elhelyezkedő dolgoknak szolgáltat egymásról információt.

- **Robot:**

Egy robot közvetlenül kapcsolatban van a terminállal és a terminálon keresztül kapcsolatban van a többi robottal.

### 3. Ágensek viselkedése

- **Terminál:**

A terminálban a futószalag mellett időközönként megjelennek csomagok, amik megjelenésekor a többi Robot ágens számára frissíti a legrövidebb utak listáját. Illetve a lerakási pontokon eltüntetni, ponttá alakítja, a csomagokat. A terminál nem rak le egy meglévő csomagra másikat, illetve a Robotra sem helyez el újabb csomag, ha éppen a futószalag mellett áll.

- **Robot:**

A Robot ágensek alapvetően a legrövidebb utak megszállottjai. Feladatuk, hogy a hozzájuk legközelebbi csomagokat felvegyék és a csomag színének megfelelő legközelebbi helyre vigyék. Ha még nem felvett csomagot látnak megindulnak felé, amint odaérnek hozzá felveszik és elindulnak vele a lerakási pontra. Ha az odaúton felvették a csomagját, amiért ment, akkor vagy újat választ magának, a helyzetének megfelelően, vagy ha nincs szabad csomag, megáll.

## A fejlesztés összefoglalása

### 1. Milyen Jason verziót használtak?

Jason 3.1 verziót használtunk, de végül nem JASON-ös megvalósításnál maradtunk.

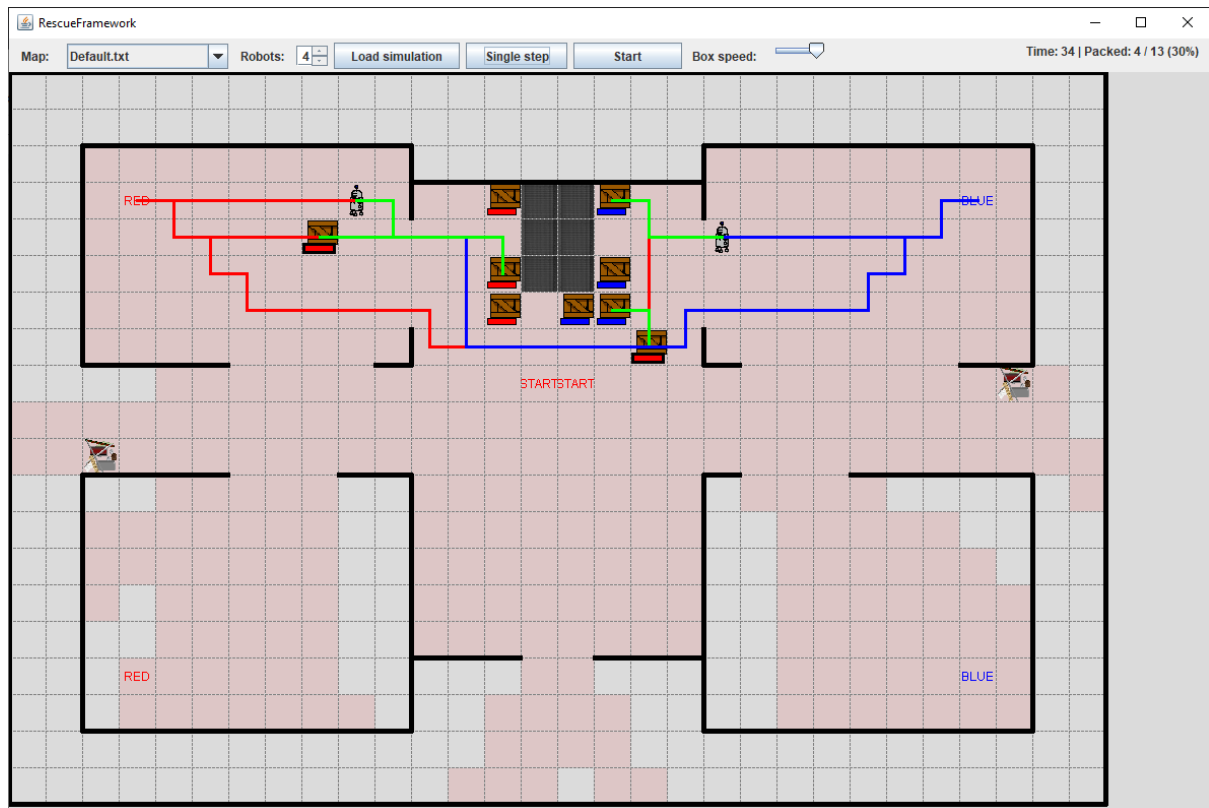
## 2. Mi készült Java szinten?

Java szinten használt fájlok:

- RobotControl.java
- AbstractRobotControl.java
- Global.java
- MainFrame.java
- PaintPanel.java
- RescueFramework.java
- Settings.java
- StepThread.java
- AStarCell.java
- AStarSearch.java
- Box.java
- Cell.java
- Floor.java
- Map.java
- Path.java
- Robot.java
- ViewLine.java
- ViewLineBreakPoint.java

# A kifejlesztett program ismertetése

## 1. Program kinézete



## 2. Kezelőfelület

A kezelőfelületen balról jobbra haladva megtalálható:

- Térképválasztó:

Az txt formátumban elkészített térképek listájából lehet választani. Ha új térképet szeretnénk hozzáadni, csak a megfelelő mappában kell elhelyezni és elérhető lesz a programból. Ez a “Load simulation” gomb lenyomása után lesz látható.

- Robot számosság állító:

A fel-le gombok, vagy a kívánt érték megadásával lehet adott mennyiségű robotot elhelyezni a térképen. Ez a “Load simulation” gomb lenyomása után lesz látható.

A START felirattól jobbra kezdi el elhelyezni a robotokat. Annyi robot elhelyezésére van lehetőségünk, amennyi egymás után ilyen módon elfér.

- “Load simulation” gomb:

A beállított értékekkel generálja a pályát és elindítja a szimulációt.

- “Single step” gomb:

A szimuláció betöltése után lehetőségünk van ezzel a gombbal egy környi lépést tennünk az ágensekkel. Segíti a viselkedésük vizsgálatát.

- “Start”/ ”Pause” / “Autostep” gomb:

A programban a pálya betöltése után ez a gomb “Start” felirattal szerepel. A gomb megnyomásakor az ágensek automatikusan működésbe lépnek és a gomb “Pause” felírra vált.

A “Pause” gomb lenyomásakor az ágensek megállnak abban az állapotban, ahogy éppen vannak. A gomb pedig “Autostep” felírra vált. Ebből az állapotból a “Single step” vagy az “Autostep” gombok megnyomásával léphetünk tovább.

Az “Autostep” gomb lehetőséget biztosít arra, hogy az ágensek ismét folyamatosan fussanak. A gomb megnyomásakor a felirat “Pause”-ra változik és funkciója a már ismertetett módon alakul.

- “Box speed” slider:

A specifikációban leírtuk, hogy lehetőség lesz a dobozok megjelenésének sebességét a GUI-n keresztül állítani. Erre szolgál ez a slider. A teljesen jobbra húzott slider a maximális doboz megjelenési értéket adja, míg a teljesen balra húzott leállítja a dobozok megjelenésének sebességét. (Ha a szalag körül már így sincs lehetőség új doboz lerakására, akkor nem kerül le újabb doboz a slider értékétől függetlenül.)

- Time X | Packed: Y/Z (N%):

A program jobb felső sarkában a program számon tartja nekünk az eltelt időt, a lepakolt/megjelent csomagok számát és ezek százalékos arányát.



## 3. Ágens működése

### 3.1. Fájlok funkcionalitása

- RobotControl.java
  - A Robot ágensek alapvetően a legrövidebb utak megszállottjai. Feladatuk, hogy a hozzájuk legközelebbi csomagokat (boxPath) felvegyék és a csomag színének megfelelő legközelebbi helyre vigyék (getShortestExitPathR vagy getShortestExitPathB). Ha még nem felvett csomagot látnak megindulnak felé, amint odaérnek hozzá felveszik és elindulnak vele a lerakási pontra. Ha az odaúton felvették a csomagját, amiért ment, akkor vagy újat választ magának, a helyzetének megfelelően, vagy ha nincs szabad csomag, megáll.

## 4. Java oldal működése

### 4.1. Fájlok funkcionalitása

- AbstractRobotControl.java
  - A robotok nem ágensként betöltött programjáért felel. Például új robot listába vétele.
- Global.java
  - A dobozok megjelenésének sebességét tartalmazza globálisan.
- MainFrame.java
  - A GUI megjelenítéséért felelős.
- PaintPanel.java
  - Panelen megjelent grafikáért felelős.
- RescueFramework.java
  - Az applikáció fő osztálya.
- Settings.java
  - Az alap beállításokat kezeli.
- StepThread.java
  - A lépések száláért felel. Ennek köszönhetően lehet léptetni a programot.
- AStarCell.java

- A cellák listájáért felelő fájl.
- AStarSearch.java
- Box.java
  - A dobozok osztálya. Ebből az osztályból lehet példányosítani őket és tulajdonságaikat állítani.
- Cell.java
  - Cellák osztálya. Itt kapják meg a tulajdonságokat.
- Floor.java
  - A padló színezéséért felelős osztály.
- Map.java
  - A pályáért felelős fájl. Ebben a fájlban található meg a “futószalag” is, ami a dobozok lepakolásának szabályait tartalmazza (incomingBox() függvény).
- Path.java
  - Két cella közötti utat reprezentál.
- Robot.java
  - A térképen lévő Robotot tartalmazza.
- ViewLine.java
  - A robot látómezejét tartalmazza.
- ViewLineBreakPoint.java
  - A robot látómezejének végét tartalmazza.

#### 4.2. Projektfájl

A RescueFramework.java

### 5. Futásidőben módosítható paraméterek

Futásidőben a dobozok megjelenésének sebességét lehet állítani a “Box speed” slider segítségével. Ez egy 0 és 100 közötti értéket állít. Ha az érték 0-án van, akkor a dobozok megjelenése megáll, 100-nál pedig maximális sebességgel jelennek meg. (Ha a szalag körül már így sincs lehetőség új doboz lerakására, akkor nem kerül le újabb doboz a slider értékétől függetlenül.)

## 6. Ágens programok rövid összefoglalása (BDI jelleggel)

- **Robot:**

A robotok alapvetően az hiszik, hogy a hozzájuk legközelebbi dobozt fel tudják venni, ezért elindulnak felé, hiszen a vágyuk, hogy a dobozt a helyére tudják vinni. Azonban, ha felveszik előlük, akkor kénytelenek új doboz után nézni. Ha pedig egy robot felvett egy dobozt, akkor a legrövidebb úton szeretné azt a legközelebbi helyére rakni a doboznak megfelelő szín szerint.

## 7. A program egészének összefoglalása (többágenses rendszer jelleggel)

A programban állítható mennyiségű robot pakol egy állítható sebességű szalag mellől két különböző dobozt, kéket és pirosat, a doboznak megfelelő helyre. A program számolja a helyre rakott dobozokat és ez alapján pontozza az ágensek összességét.

## 8. Link a videóhoz

<https://www.youtube.com/watch?v=qAU7BqkBoNI>