

Intelligens elosztott rendszerek (VIMIA C02)

Házi feladat:

Intelligens közúti kereszteződések intelligens járművekkel

Készítette:

Horváth András
(NK: YKBGW3)

2017/2018 II. félév

Tartalomjegyzék

Feladat ismertetése.....	3
Megoldás.....	4
Lámpa	4
Jármű	4
Implementáció	5
Ágens kódok	6

Feladat ismertetése

A feladatban megoldandó probléma helyszíne egy nem túl bonyolult közúti kereszteződés. A kereszteződésen intelligens járművek és gyalogosok közlekednek. Ezt az 1. ábra mutatja.

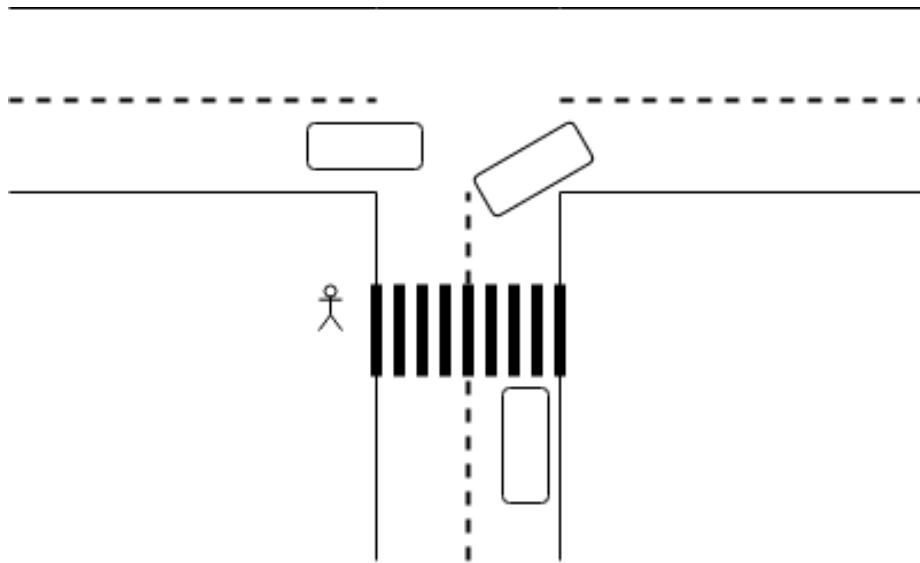
Az intelligens járművek a kereszteződésben összefonódottan haladnak át. Mindegyik szeretne a lehető leggyorsabban átjutni és természetesen az ütközést mindenképpen elkerülni. Az ehhez szükséges információkat megosztják egymással.

Ilyen információ:

- a jármű sebessége, és távolsága kereszteződéstől,
- a maximális gyorsulás és lassulás, amit végre tud hajtani
- a jármű hossza (egyszerűsítés: egyforma hosszúak)

A járművekről feltételezem hogy sokkal gyorsabban számolnak és kommunikálnak mint haladnak, ezért megtervezhetik az áthaladást a pillanatnyi állapot alapján, figyelmen kívül hagyva a tervezés ideje alatt megtett utat.

A kereszteződés intelligenciája abban áll, hogy érzékeli az átkelni szándékozó gyalogosokat, és ennek ismeretében kommunikál a járművekkel. A gyalogosok miatt - minthogy jóval lassabbak a személygépjárműveknél - nem célszerű sűrűn megállítani a járműveket, ezért a kereszteződés megvárja amíg többen összegyűlnek. Ilyenkor a gyalogosokkal párhuzamos forgalom haladhat.



1. ábra Egyszerű kereszteződés

Megoldás

A környezetben lévő ágensek a jármű (Car.asl) és a lámpa (Lamp.asl).

Lámpa

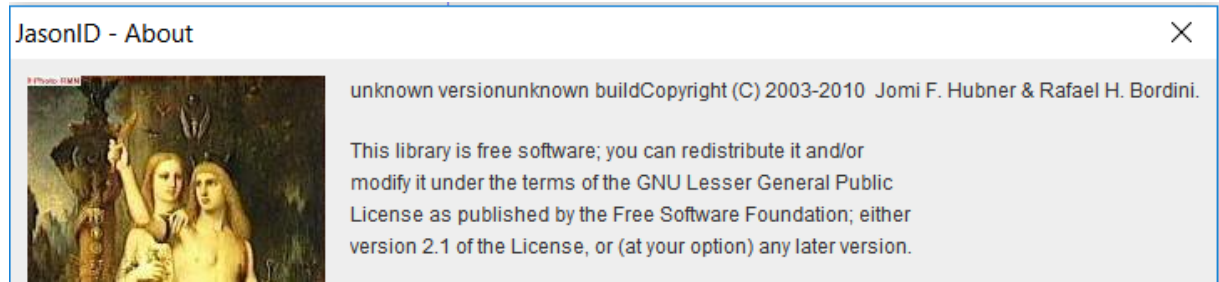
A jelzőlámpát a környezete értesíti az újonnan érkező gyalogosokról. Ekkor a lámpa frissíti a tudásbázisát a jelenleg várakozó gyalogosok számáról. Amikor egy jármű jelzi, hogy közeledik felé és elég sok a várakozó gyalogos, akkor egy `.send(Name, tell, collision (lamp))` üzenetet küld válaszul.

Jármű

A jármű a környezetéből tudja meg a az induláskor aktuális állapotát, és erről a többi ágénst a környezetben tájékoztatja. Egy ilyen tájékoztató üzenet fogadásakor kiszámolja hogy a most kapott információ veszélyes-e magára nézve. Amennyiben igen, akkor erről a választ küld a bejelentőnek.

Implementáció

A programhoz használt Jason verzió: 2.2a (a sourceforge szerint)



Javában készítettem egy hálómodellt a MarsRobotok példa alapján, de a feladatom vizualizálására nem nagyon alkalmas.

A java környezetben egy TestCase ösosztályból leszármazva több tesztet definiálható és a konzolon megjelenik az ágensek cselekedetei.

Az agensek viselkedése teljesen ASL nyelvben készült, egy cselekvést implementáltam Java nyelven. A járművek így ellenőrizni tudják, hogy egy másik mozgó jármű zavarja-e őket. Ha igen, akkor a tudásbázisába bekerül ez az információ. Ez azt modellezi hogy egy másik fedélzeti rendszer végzi el az ehhez szükséges számítást és az eredményről értesíti a működtetést végző ágenszt.

Ágens kódok

Car.asl

```
// Agent car in project Crossroad.mas2j

/* Initial beliefs and rules */

/* Initial goals */

/* Plans */

+test(X) : true <-
  -test(X);
  .my_name(Name);
  .print("im tested, my name is ", Name).

+start(X, Y, SPEEDX, SPEEDY, GOALX, GOALY): true <-
  -start(X, Y, SPEEDX, SPEEDY, GOALX, GOALY);
  +pos(X, Y);
  +speed(SPEEDX, SPEEDY);
  +moveto(GOALX, GOALY);
  .send(lamp, tell, newPed);
  !tellMyState.

+!tellMyState : true <-
  .my_name(Name);
  ?pos(X, Y);
  ?speed(SPEEDX, SPEEDY);
  ?moveto(GOALX, GOALY);
  .broadcast(tell, moving(Name, X, Y, SPEEDX, SPEEDY, GOALX, GOALY)).

+moving(Name, X, Y, SPEEDX, SPEEDY, GOALX, GOALY)[source(Name)] : pos(_, _)
  <-
  .print("checking safety")
  calcIfSafe.

+collision(Name) <-
  .print("WARNING collision with: ", Name);
  .my_name(ME);
  .send(Name, tell, collision(ME)).

// Agent car in project Crossroad.mas2j
```

Lamp.asl:

```
// Agent lamp in project Crossroad.mas2j

/* Initial beliefs and rules */
// no pedestrian waiting at the beginning
numPed(0).

/* Initial goals */

/* Plans */

// stop the trafic if enough pedestrian is present
```

Intelligens közúti kereszteződések intelligens járművekkel

```
+moving(Name, X, Y, SPEEDX, SPEEDY, GOALX, GOALY)[source(Name)]:  
  numPed(NUM) & NUM>2  
  <-  
  .send(Name, tell, collision(lamp)).  
  
// not enough pedestrian to stop the traffic  
+moving(Name, X, Y, SPEEDX, SPEEDY, GOALX, GOALY)[source(Name)]  
  <- true.  
  
+newPed <-  
  .print("new pedestrian arrived");  
  !updateCnt;  
  -newPed[source(_)].  
  
+!updateCnt : true <-  
  ?numPed(X);  
  -numPed(X);  
  +numPed(X+1);  
  -newPed.  
// Agent lamp in project Crossroad.mas2j
```