



## **PET felhasználói kézikönyv**

*Részvételi kiterjesztés  
Participatory Extension*

*Készítette:  
AITIA International, Inc.  
Az ELTE-IKKK részére.*



*A projekt az EU társfinanszírozásával, az Európa Terv  
keretében valósul meg.*

*Budapest, 2007*

## Tartalomjegyzék

PET felhasználói kézikönyv	1
1 Bevezetés	3
1.1 Ágens-alapú modellezés	3
1.2 MASS	3
1.3 PET	4
1.4 Cél	4
1.5 Rendszerkövetelmények	4
1.5.1 Minimum Rendszerkövetelmények	4
1.5.2 Optimális Rendszerkövetelmények	4
1.6 Szerepek	4
1.7 Belépés	4
1.7.1 Sikertelen bejelentkezés	6
1.8 Felhasználói navigáció	6
2 Modellek és szimulációk	7
2.1 Szimulációk létrehozása	7
2.2 Csatlakozás szimulációkhoz	7
2.3 Felhasználói beállítások (szimulációkra)	8
2.4 Szimuláció vezérlés	10
2.5 Ágensek vezérlése	10
2.5.1 Ágens elengedése	12
2.5.2 Szimulációk reprodukálása	12
3 Befejezés	13
4 A Függelék - Modell leírások	14
4.1 Fire (tűz)	14
4.2 Hunter (vadász)	14
4.3 Zerosum (nullösszeg)	14
4.3.1 Matematikai modell	15
4.3.2 Rövid analízis	15
4.3.3 Ágens stratégiák	15
4.3.4 Hivatkozások a Zerosum modellhez	17

# 1 Bevezetés

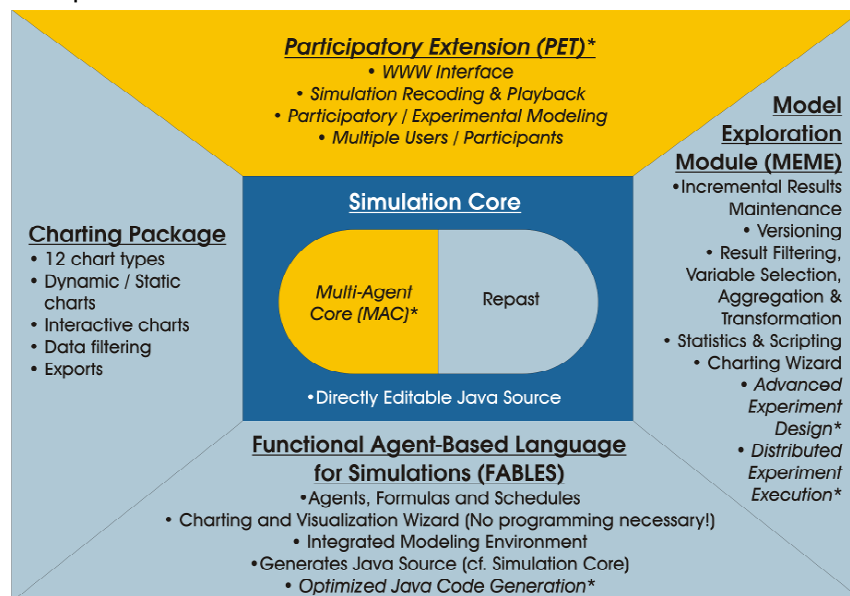
## 1.1 Ágens-alapú modellezés

Az ágens-alapú modellezés a számítógépes szimulációk egy - komplex társadalmi rendszerek modellezésére különösen alkalmas - új ága. Alapvetése az, hogy az egyént modellezzük, tökéletlenségeivel (pl. korlátozott kognitív és számítási képességek), egyéni jellegzetességeivel és egyedi interakcióival együtt. A modell tehát "alulról felfelé" épül - elsősorban a mikro szabályokra koncentrálva, de a makró jelenségek kialakulását kutatva. A részvételi szimuláció, mint az ágens-alapú szimuláció alfaja egy olyan metodológia, amely az emberi szereplők és a mesterséges ágensek együttműködésére alapoz. Ezek a megoldások a képzési és döntés-támogatási területeken igen hasznosak.

## 1.2 MASS

A Multi-Agent Simulation Suite (MASS) egy szoftvercsomag, amely lehetővé teszi a felhasználó számára, hogy az ágens-alapú modellezés megoldásait változatos területeken alkalmazza anélkül, hogy komoly programozási ismereteket kelljen elsajátítania.

A programcsomag négy, egy ún. szimulációs mag köré szerveződő alkalmazásból áll össze. A MASS rendelkezik egy saját maggal (ez a MAC), illetve képes futni Repast alapokon is. A több szimulációs magon való futtathatóság biztosítja, hogy a modellek mag-függetlenek legyenek, így a használható magok számát a jövőben bővíteni kívánjuk. A Functional Agent-Based Language for Simulation (FABLES) egy olyan programozási nyelv és modellezési környezet, amely kifejezetten ágens-alapú modellek fejlesztését szolgálja. A Model Exploration Module (MEME) paraméter terek bejárását, a kinyert adatok feldolgozását és megjelenítését hivatott támogatni. A Participatory Extension (PET) egy opcionális web-alapú környezet multi-ágens és részvételi szimulációk futtatásához. A MASS negyedik része, a Vizualizációs Csomag, nem jelenik meg önálló programként, a többi szoftverben használt grafikonok és vizualizációk implementációit tartalmazza.



1. ábra - Multi-Agent Simulation Suite

## **1.3 PET**

A PET tehát egy webes környezet modellek és multi-ágens szimulációk futtatására és adminisztrálására. A magja, lehet az eredeti Multi-Agent Core (Multi-Ágens Mag, MAC), vagy a Repast. A PET, része a Multi-Agent Simulation Suite-nek (MASS).

## **1.4 Cél**

Ezt a használati utasítást a PET-et használó végfelhasználóknak állítottuk össze, ezért, nincs szó benne programozási modellekről vagy a PET adminisztrációjáról. Ezekről részletes leírás található a PET Modell írók kézikönyvében és a PET Adminisztrátori kézikönyvben.

## **1.5 Rendszerkövetelmények**

### **1.5.1 Minimum Rendszerkövetelmények**

A PET futtatásához Java 1.5 kompatibilis platform vagy későbbi (Pentium 166MHz vagy gyorsabb, minimum 64 MB RAM) szükséges.

### **1.5.2 Optimális Rendszerkövetelmények**

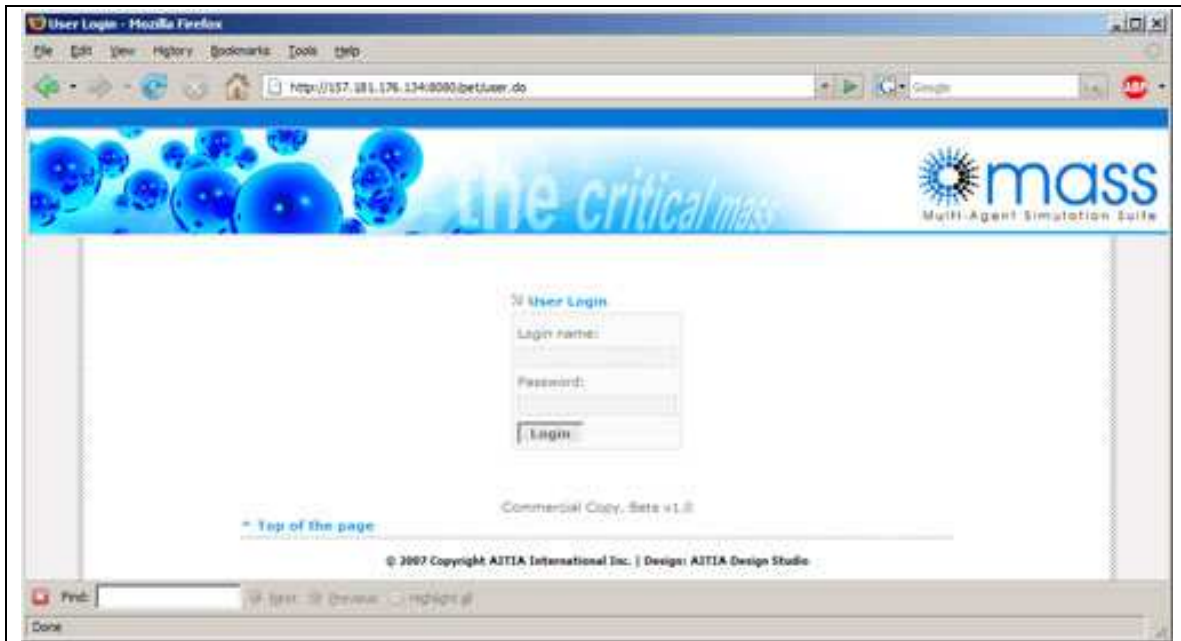
Pentium 1000MHz ekvivalens vagy gyorsabb és 256 MB RAM.

## **1.6 Szerepek**

Alapvetően kétfajta felhasználója van a PET-nek; végfelhasználók, akik futtatják és részt vesznek a szimulációkban és az adminisztrátorok, akik modelleket hoznak létre és adminisztrálják a végfelhasználókat. Ez a kézikönyv az első csoportnak készült, míg az adminisztratív kérdéseket a PET Adminisztrátori kézikönyv tárgyalja.

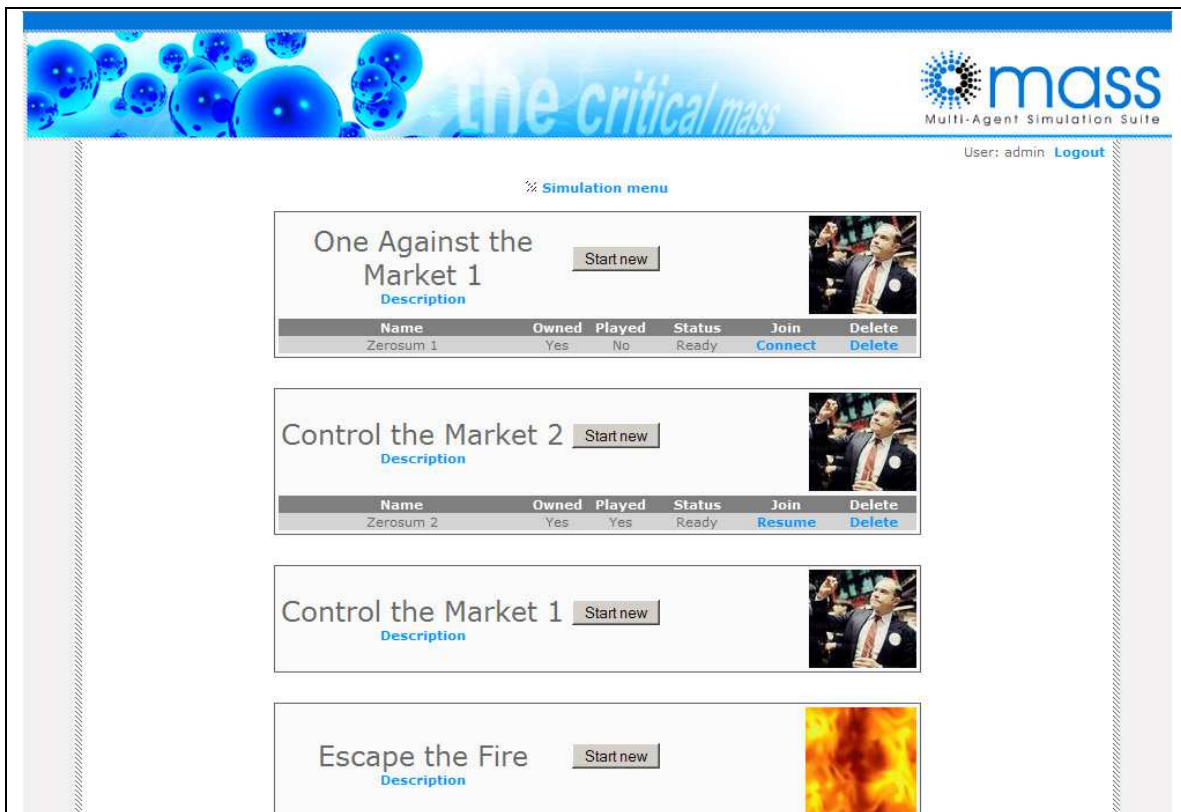
## **1.7 Belépés**

A PET indításához nyissa meg böngészőjét, – lehetőleg Mozilla Firefox vagy MS Internet Explorer– és írja be az adminisztrátortól kapott címet. A PET Web-es interfészt használ, és be kell jelentkeznie az adminisztrátortól kapott felhasználó névvel és jelszóval, mint bármelyik internetes oldalra. Az alábbi képen látható a bejelentkezési oldal:



## 2. ábra Belépés

Ha a fenti képhez hasonló bejelentkezési oldalt lát, írja be felhasználó nevét és jelszavát a megfelelő mezőkbe, majd kattintson a **Login** gombra. A hitelesítés megtörténte után a felhasználói főoldalnak kell megjelennie. (lásd lent)



## 3. ábra - Főoldal - Szimulációs menü

### 1.7.1 Sikertelen bejelentkezés

Amennyiben elgépelte a felhasználónevet vagy a jelszót, a következő üzenetet láthatja:



#### 4. ábra – Sikertelen bejelentkezés

Bármilyen más hiba vagy hibaüzenet, ami a bejelentkezés során merül fel más eredetű. Ezekben az esetekben a rendszergazdával kell felvenni a kapcsolatot!

## 1.8 Felhasználói navigáció

A felhasználó-oldali navigációt a szimulációk köré szerveztük. Minden meglévő és rendelkezésre álló szimuláció szerepel a főoldalon (3. ábra), amely a bejelentkezés után jelenik meg. A felhasználó csatlakozhat a szimulációkhoz vagy újakat hozhat létre, miután csatlakozott egy szimulációhoz, olvashatja a teljes leírást és hozzáférhet minden más opcióhoz.



#### 5. ábra – Navigációs sáv a szimulációs vezérlőoldalon

A szimulációs vezérlőoldalon (5. ábra) a rendelkezésre álló opciók és parancsok háromszintes hierarchiában láthatók a felhasználói oldalon. A szimulációs vezérlőgombok és parancsok a szimulációs vezérlősávon találhatóak. (a fenti képen a sötétszürke sáv)

## 2 Modellek és szimulációk

A PET-et a felhasználók úgynevezett modellcsaládok (Model Family), a modellek (Model) és szimulációk (Simulation) hierarchiáján keresztül használják. A modelleket az adminisztrátorok állítják be a modell családokból úgy, hogy megadják az ágensek számát a modell bizonyos verziójában és egyéb konfigurációkat is meghatároznak. A szimuláció egy rendelkezésre álló modell verziója, mely futtatható, ismételhető, és részt lehet benne venni. (vagyis egy modellhez számos szimuláció tartozhat.) A végfelhasználók a modelleket és a szimulációkat modellek szerint csoportosítva látják (lásd 3. ábra). A végfelhasználók úgy érhetik el a rendelkezésre álló szimulációkat, hogy a 'Simulations' linkre kattintanak a felhasználói felületen. (5. ábra)

### 2.1 Szimulációk létrehozása

A végfelhasználók a rendelkezésükre álló modellekből tudnak új szimulációkat létrehozni. A szimuláció tulajdonosa (**owner**), az azt létrehozó felhasználó lesz. Szimulációt létrehozni a rendelkezésre álló modellek listájában lévő „Start New” (lásd lent) gombra kattintva lehet,.



6. ábra – szimuláció indítása

Fontos tudni, hogy az adminisztrátor is tud szimulációt létrehozni. A szimulációt létrehozó felhasználó automatikusan a szimuláció tulajdonosává válik. Az adminisztrátorokon kívül csak a tulajdonos törölhet egy szimulációt.

### 2.2 Csatlakozás szimulációkhoz

A felhasználók egyaránt részt vehetnek olyan szimulációkban, melyeknek ők vagy más felhasználók a tulajdonosai, amennyiben a szimulációban vannak rendelkezésre álló, vezérelhető ágensek és/vagy a felhasználó jogosult az adott szimuláció megtekintésére.



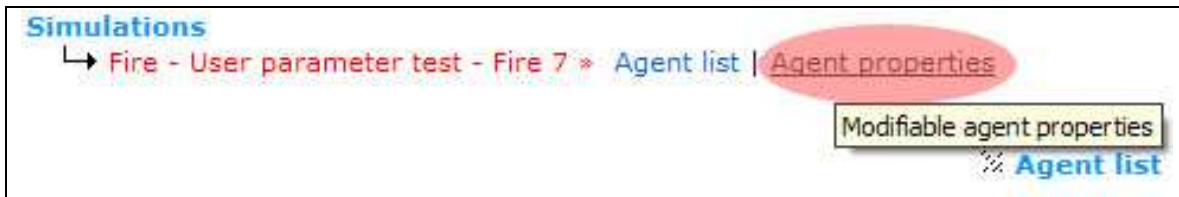
7. ábra - Csatlakozás/Kapcsolódás

A végfelhasználók úgy csatlakozhatnak egy szimulációhoz, hogy a főoldalon rákattintanak a **Connect** szóra – ha van - az adott modell sorában. Ebben a sorban

alapinformációk található; a szimuláció neve, hogy a felhasználó tulajdonosa-e a szimulációnak, lejátszották-e már a szimulációt, a szimuláció státusza, az előbb említett **Connect** parancs és a **Delete** parancs, ami kizárólag akkor aktív, ha a felhasználó a tulajdonosa a szimulációnak. (lásd fent)

## 2.3 Felhasználói beállítások (szimulációkra)

Bár a PET felhasználói oldalán megjelenő összes szimuláció „indításra kész”, néhány modellben a felhasználó is változtathat az ágensek tulajdonság értékein. Ezeket a modelleket az adminisztrátorok alapértelmezett ágens tulajdonságokkal állítják össze, melyeket a szimulációt elindító felhasználó megváltoztathat. A módosítható beállításokkal rendelkező szimulációk esetében van egy extra opció – *ágens tulajdonságok* – a szimulációs vezérlőoldalon. (lásd lent)




### 8. ábra – Módosítható tulajdonság értékekkel rendelkező szimuláció


A *módosítható ágens tulajdonságok* oldalon (lásd 8. ábra) az összes módosítható tulajdonságokkal rendelkező ágens szerepel a listában. Egy csoportban szerepelnek azok az ágensek melyeknek ugyanazon tulajdonsága módosítható. Az összes ágens tulajdonság listázható, ha kipipáljuk a *Csak olvasható tulajdonságokat*.

**Modifiable agent properties**


Show read only properties

 **Fire** Agent id: 586


Property	Class	Value	Variety
Increment	java.lang.Double	<input type="text" value="1000.0"/>	1

 **Person**   instances.

Property	Class	Value	Variety
Eyeshot	java.lang.Double	<input type="text" value="30.0"/>	1

 **Person** [List instances](#)   instances.

Property	Class	Value	Variety
Eyeshot	java.lang.Double	<input type="text" value="30.0"/>	1
Location X	java.lang.Integer	<input type="text"/>	2
Location Y	java.lang.Integer	<input type="text"/>	2


 **World** Agent id: 585

Property	Class	Value	Variety
Heat spread on walls	java.lang.Double	<input type="text" value="0.95"/>	1

**9. ábra – Módosítható ágenstulajdonság lista**

Ahhoz, hogy egy tulajdonságot megváltoztassunk, a kívánt értéket be kell írni az érték mezőbe és rákliccelni az **Update** gombra. (lásd lent, ahol a növekedést 1000.0-re állítottuk)

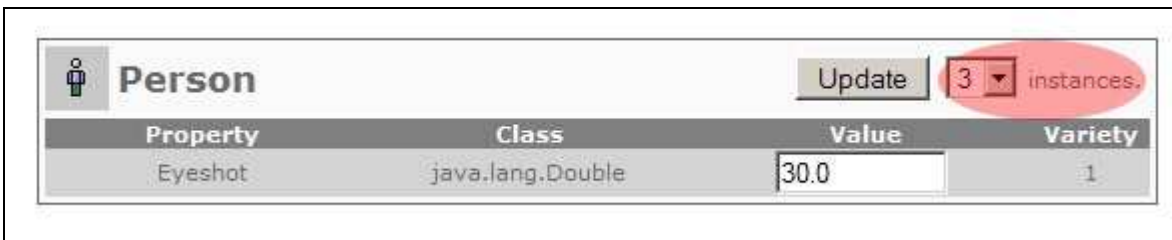
Show read only properties

 **Fire** Agent id: 586

Property	Class	Value	Variety
Increment	java.lang.Double	<input type="text" value="1000.0"/>	1

**10. ábra - Kiklikeljen az *Update* gombra a tulajdonság-érték megváltoztatásához**

Ha több ágens van ugyanazzal a módosítható tulajdonsággal, az **Update** gomb oldalán található legördülő listából lehet kiválasztani, hogy hány esetben változzon meg az értékszám. (11. ábra).



**11. ábra - Több ágens ugyanazzal a módosítható tulajdonsággal**

Amikor több ágens van többféle tulajdonsággal, lehetséges mindegyiknek külön az összes tulajdonság értékét megadni és kilistázni, ha a '*List instances*' linkre kattintunk (12. ábra).



**12. ábra – Két ágens ugyanazzal a három módosítható tulajdonsággal.**

A szárosság (*Variety*) oszlop az ágensekben előforduló adott tulajdonság különböző értékeinek számát mutatja.

## 2.4 Szimuláció vezérlés

A szimuláció kiválasztásával a felhasználó a szimulációt vezérlő oldalra kerül. Innen a felhasználó indíthatja, megállíthatja, újrazedheti és megismételheti a szimulációt, valamint megfigyelheti a vizualizációkat, vezérelheti az ágenseit a „résztvételi modellekben”.



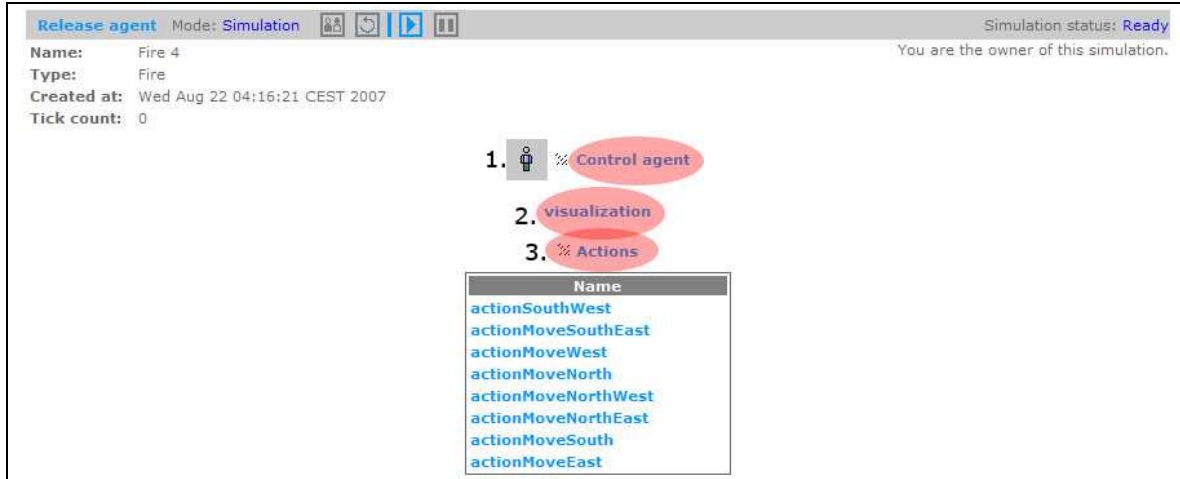
**13. ábra – Szimuláció vezérlők**

Az első két vezérlőgomb, a felső ábrán számmal jelölve, a következő: *Szimuláció mód* (1) *Visszajátszó mód* (2). A fenti képen a szimuláció még nem lett elindítva. Miután elindítottunk a 3-as gomb megnyomásával, az 1-es gomb kék lesz (aktív), ami jelzi, hogy szimulációs módban van. Ha megállítják, (4.) a szimulációt *Visszajátszó mód*ba lehet kapcsolni a 2-es gomb megnyomásával, a visszajátszás indításához pedig a *Play* (3.) gombra kell klikkelni. Szimulációs módba úgy lehet visszaváltani, hogy ismét megállítjuk, majd az 1-es gombra kattintunk.

## 2.5 Ágensek vezérlése

A „résztvételi modellekben” a felhasználók az ágensek egy részét vagy az mindet vezérelhetik a többit a számítógép irányítja. Ahhoz, hogy egy ágens „megfogjunk” a *Szimulációs menüben* a *Connect-re* kell klikkelni. (6). Ha csak egy vezérelhető ágens van a „résztvételi modellben” akkor a *PET* a felhasználót automatikusan az adott

ágenst vezérlő oldalára irányítja amikor a felhasználó csatlakozik a szimulációhoz. (lásd lent)



#### 14. ábra – Ágens vezérlés

A fenti képen a vizualizáció nincsen beágyazva a ágens vezérlő oldalba. Ez inkább a kivétel mint a szabály, de ez esetben a **Visualization**-re kattintva egy 3D-s vizualizációs alkalmazás indul el.

Az **Actions** alatt fel van sorolva az összes akció amire az ágens képes a modellben. A fenti képen ezek mozgással kapcsolatosak, de lehet bármi amit a modell alkotója meghatároz. Fontos megemlíteni, hogy amikor a felhasználó rákattint egy vezérlő linkre, az utasítás nem hajtódik végre egyből, hanem csak egy pufferbe tárolódik. Valójában akkor hajtódik végre, amikor az ágens a rendszer által aktiválásra kerül. Normál esetben ez a szimuláció minden körében egyszer történik meg. A puffer egyszerre csak egy utasítást tud tárolni. Ezért ha a puffer nem üres miközben egy új utasítás érkezik, a régi utasítás felülíródik az újjal, és a régi sohasem hajtódik végre.

Ha több mint egy vezérelhető ágens van a szimulációban, a felhasználó a (vezérelhető) ágens lista oldalára kerül (lásd 14. ábra) ahol ki lehet választani a megtekinteni vagy vezérelni kívánt ágenseket.



#### 15. ábra - 1. Ágens megtekintése; 2. Ágens vezérlése

Fontos, hogy a fenti ábrán a kiemelt ikonok kékek, ami azt mutatja, hogy a 9, 10, 11 ágensek vezérelhetők.

### 2.5.1 Ágens elengedése

Az ágens elengedéséhez – hogy az önállóan működjön vagy más felhasználók vezéreljék – egyszerűen kattintson a *Release agent* parancsra az Ágens vezérlő lapon.



16. ábra – Ágens elengedése

### 2.5.2 Szimulációk reprodukálása

A PET-ben minden szimuláció akárhányszor újrajátszható. Ez azért lehetséges, mert minden felhasználói akció mentésre kerül, így a szimuláció visszajátszásakor az eredeti futás reprodukálható.

## 3 Befejezés

A PET egy olyan szimulációs szoftver, ami minőségi szolgáltatásokat nyújt a modellezőknek részvételi szimulációk készítéséhez, amelyekben egyének és szoftver ágensek működhetnek együtt. A felhasználói felület könnyen kezelhető, intuitív, így a felhasználók hamar beletanulnak a szoftver kezelésébe. A felhasználói interfész és az adminisztrátori interfész jól elkülönül, ami megkönnyíti a program kezelését. A dokumentum a végfelhasználóknak fejlesztett felületet mutatja be. Ez a felület lehetővé teszi a felhasználók számára, hogy létrehozzanak, vezéreljenek, és részt vegyenek a szimulációkban, irányítsák az ágenseket és állítsák az azok tulajdonságait futás közben.

## 4 A Függelék - Modell leírások

Ez a fejezet azokról a modellcsaládokról és az azokból létrehozott modellekről ad leírást amelyek alapértelmezés szerint bekerülnek a rendszerbe telepítéskor.

### 4.1 Fire (tűz)

Ez a modell egy egyszerű példája a vészhelyzetek szimulációjának. A modell „világa” egy épület, amiben falakkal elválasztott szobák vannak. Hirtelen tűz üt ki az épület egyik pontján. A tűz megadott ütemben terjed, de a falak kissé lelassítják. Az épületben lévők megpróbálnak kimenekülni a kijáraton. Néhányuknak ez sikerül, míg néhányuknak nem. A modellhez 2D és 3D megjelenítést is kínálunk, mind az ágensek, mind a világ részére. A szimulált ágensek nem ismerik a kijáratok helyét és korlátozott a látóképességük. A viselkedésük úgy van programozva, hogy kövessék egymást, menjenek el a tűztől és derítsék fel a környezetüket.

Két némiképp eltérő futtatásra kész modell tartozik a PET-hez adott Fire modellcsaládhoz:

1. *Watch the Fire (nézd a tüzet):* Az épület felülnézeti, kétdimenziós megjelenítésében nézhetjük, ahogy a tűz terjed és az ágensek próbálnak kimenekülni.
2. *Escape the Fire (menekülj ki a tűzből):* Az egyik ágens szemszögéből láthatjuk az ő menekülését az épületből. Ehhez háromdimenziós megjelenítést használ!

### 4.2 Hunter (vadász)

A vadász modell egy egyszerű modell kétféle ágenssel (szigorúan véve háromféleképp, mert a PET a teret is ágensnek tekinti). Néhány ágens *vadász*, míg mások *prédák*. Egy kétdimenziós periodikus térben élnek. Ahogy a nevükből is kitűnik, a vadászok megpróbálják elkapni a prédát, míg azok megpróbálnak elmenekülni előlük. Ha egy vadász elkapja prédáját, a préda eltűnik, és vagy egy másik véletlenszerű helyen bukkan fel, vagy eltűnik a világból, ez a modell beállításától függ. A vadászok és a prédák bizonyos tekintetben nagyon hasonló viselkedést mutatnak, mindketten a (legközelebbi) ellentétes típusú ágenseket keresik, de a vadászok a préda felé mozdulnak, míg a prédák igyekeznek eltávolodni a vadászoktól.

A vadász modell programozásáról részletesen lásd a *PET modell írók kézikönyve dokumentumot!*

Két némiképp eltérő futtatásra kész modell tartozik a PET-hez adott Hunter modellcsaládhoz:

1. *Watch the Hunt (nézd a vadászatot):* Meg lehet figyelni a vadászatot.
2. *Don't Be a Prey! (Ne légy préda!):* Az egyik prédát kell irányítani. Kerüld el a vadászokat!

### 4.3 Zerosum (nullösszeg)

A Zerosum (Huber és Kirchler; Yoneda et al) modell egy egyszerű, információ alapuló null-összegű játék a határidős piacon. Rámutat arra a tényre, hogy a leginkább és a legkevésbé informált játékosok többet keresnek, mint a közepesen informáltak.

### 4.3.1 Matematikai modell

A játékot  $2M + 1$  játékos játssza.

1. Minden kör elején az  $i$ -edik játékos ( $0 \leq i \leq 2M$ ) közli a küszöbárát,  $R_i$ -t, ezzel egy ilyen szerződést köt: ő megvesz (elad) egy egységnyi terméket, ha  $R_i$  magasabb (alacsonyabb), mint a tényleges ár,  $P$ .
2. Az árverésvezető begyűjti a küszöbárakat és kiválasztja közülük a mediánt, amit egyenlővé tesz  $P$ -vel, így a kereslet megegyezik a kínálattal a határidős piacon.
3. Miután bezárt a piac, felfedik az áru valós értékét,  $V$ -t.  $V$ -t exogén módon számolják  $2M$  darab bináris valószínűségi változó összegéből. Ezek a változók:  $X_1, X_2, \dots$ , és  $X_{2M}$  függetlenek, 0 és 1 lehetséges értékekkel, amiket 0,5 valószínűséggel vesznek fel minden körben.
4. Azok, akik vásároltak (eladtak) árut a határidős piacon, azoknak el kell adniuk (meg kell vásárolniuk) azt a készárupiacon a valós  $V$  értékén, hogy bezárhassák a számlájukat. Így minden játékos nyeresége kiderül, mikor  $V$  értékét felfedik. Azok a játékosok, akiknek  $P < R_i$ , azok megveszik az árut a határidős piacon  $P$ -ért és eladják a készárupiacon  $V$ -ért, tehát mindegyikük  $V - P$  nyereséget termel, azok a játékosok, akiknél  $R_i < P$ , azok eladják az árut a határidős piacon  $P$ -ért és megveszik a készárupiacon  $V$ -ért, tehát  $P - V$  nyereséget termelnek. Mondani sem kell, hogy a negatív nyereség a veszteség, és az összes nyereség összege mindig nulla:  $\Sigma(P - V) + \Sigma(V - P) = 0$ .

Azt feltételezzük, hogy mielőtt az  $i$ -edik játékos meghatározza  $R_i$  -t, azelőtt ő látja az  $X_1, X_2, \dots$ , és  $X_i$  változók értékét (a 0-adik játékos egyet sem tud belőlük). Látszólag az  $i$ -edik játékosnak előnye van a  $j$ -edikkel szemben ( $j < i$ ), mert az előbbi tudja, amit az utóbbi tud és még azt is, amit az utóbbi nem tud ( $X_{j+1}, X_{j+1}, \dots$ , and  $X_i$ ). Így  $i$ -re, mint „információs szintre” hivatkozunk.

### 4.3.2 Rövid analízis

A  $2M$ -edik játékos, vagyis a legjobban informált, aki lát minden  $X_i$ -t, valószínűleg pozitív nyereséget termel hosszú távon. Valójában választhatja  $\Sigma X_i$  -t, mint a küszöbárát,  $R_{2M}$ -et, és mint könnyen belátható 0 nyereséget termel, ha  $V=P$ , és pozitív nyereséget abban a sokkal inkább valószínű esetben, hogy  $P \neq V$ . Viszont, a 0. játékos, a legkevésbé informált, aki nem tud semmit az  $X_i$  -kről, nem biztos, hogy negatív nyereséget könyvelhet el hosszú távon. Ugyanis mindig választhat olyan magas (alacsony)  $R_0$ -át, hogy minden körben vásárol (elad) a határidős piacon. Egy másik gondolatmenet szerint akár véletlenszerűen is választhatja  $R_0$ -át minden körben. Mindkét esetben hosszú távon 0 a nyereségének várható értéke, mert a  $P < V$  és  $V < P$  események azonos valószínűséggel fordulnak elő.

Ezekből a speciális körülményekből egy kérdés vetődik fel, jelesen abból a körülményből, hogy a játék null-összegű. Ha a leginkább informált játékos pozitív nyereséget könyvelhet el, míg a legkevésbé informált nulla nyereséget, akkor néhány közepesen informált játékosnak veszteséget kell elszenvednie. Mégis különös, hogy az a játékos, aki jobban informált, kevesebbet keressen a játékon, mint a nála kevésbé informált.

### 4.3.3 Ágens stratégiák

Az ágensek a következő két egyszerű stratégia valamelyikét játsszák:

A várható-érték stratégia:

$$R_i = (\Sigma X_i) + 0.5 \times (2M - i)$$

Itt a jobboldal azt az értéket testesíti meg, amennyire  $V$  értékét becsli az  $i$ -edik játékos; az összeg első tagja a játékos által ismert valószínűségi változók értékeinek összege, míg a második tag a nem ismert változók összegének várható értéke.

Az extrém stratégia:

$R_i = (\sum X_i) 0,5$  valószínűséggel

$R_i = (\sum X_i) + (2M - i) 0,5$  valószínűséggel

Itt  $(\sum X_i)$  az  $i$ -edik játékos által az ő tudása alapján a  $V$ -re becsült minimum értéket jelenti, míg a  $(\sum X_i) + (2M - i)$  a maximális értéket.

Négy némiképp eltérő futtatásra kész modell tartozik a PET-hez adott Zerosum modellcsaládhoz:

1. *Egyedül a piac ellen 1*: Az összes ágenszt a számítógép vezérli, a felhasználó egy megfigyelő a piacon. Ez egy „egyedül mindenki ellen” beállítás, egy kivétellel az összes ágens a várható-érték stratégiát játssza.
2. *Egyedül a piac ellen 2*: Ez gyakorlatilag ugyanaz, mint az Egyedül a piac ellen 1, de a stratégiák fel vannak cserélve, itt a legtöbb ágens az extrém stratégiát játssza.
3. *Irányítsd a piacot 1*: A felhasználók irányíthatnak 0, 10, 20, 30, 40, 50, 70, 80 vagy 90 ágenszt a 100-ból. Az összes többi ágens a várható-érték stratégiát játssza.
4. *Irányítsd a piacot 2*: Ez gyakorlatilag ugyanaz, mint az Irányítsd a piacot 1, de a mesterséges ágensek az extrém stratégiát játsszák.

Az utóbbi két esetben a felhasználók irányíthatnak ágenseket, így befolyásolják az árakat a piacon.

#### **4.3.4 Hivatkozások a Zerosum modellhez**

Jürgen Huber and Michael Kirchler “The Value of Information in Markets with Heterogeneously Informed Traders – and Experimental and a Simulation Approach”, presented at 9th Workshop on Economics and Heterogeneous Interacting Agents (WEHIA2004), Kyoto University, Kyoto, Japan

Hiroyasu Yoneda, Gen Masumoto and Sobei H. Oda: „Marginal Contribution of Information to Profit in a Zero-sum Game”, presented at *EES2004: Experiments in Economic Sciences - New Approaches to Solving Real-world Problems*