Debreceni Egyetem
Informatikai Kar

A kétszemélyes játékok interaktív és geometriai reprezentációi

Témavezetők:

Prof. Dr. Terdik György
egyetemi tanár

Dr. Habil. Máth János
egyetemi docens

Készítette:

Münnich Ákos
Programtervező Informatikus B.Sc.

Debrecen

2010

Tartalomjegyzék

Köszönetnyilvánítás

Szeretnék köszönetet mondani Prof. Dr. Terdik Györgynek és Dr. Habil. Máth Jánosnak, akik elvállalták szakdolgozatom témavezetését, és tanácsaikkal segítették munkámat.

1. Bevezetés

A játékelmélethez kapcsolható munkák már a tizennyolcadik században megjelentek, de az első igazán fontos és átfogó tudományos értékű munka Neumann és Morgenstern (1944) alapműve. A játékelmélet szélesebb körű elterjedését nagyban segítette Luce and Raiffa (1957) könyve, ami kritikusan mutatja be a játékelmélet előnyeit és korlátait, esetenként hátrányait illetve tisztázatlan fogalmait. A játékelmélet mára rendkívül szerteágazó szakirodalommal és alkalmazási területekkel bír, aminek szisztematikus áttekintése meghaladja ezen szakdolgozat kereteit. Ha játékelmélet szakirodalmát, a teljesség igénye nélkül áttekintjük, észrevehetjük, hogy a játékelmélet legelfogadottabban és ennek megfelelően a legjelentősebben a közgazdaságtudomány keretén belül fejlődött (persze a matematikai apparátus egyidejű fejlesztésével), de a társadalomtudományi és politikatudományi igények a pszichológia jelentős bevonódását is magával vonta.

Érdemes itt megjegyezni, hogy a játékelmélet természetes módon magába foglalja a „valódi" játékok matematikai leírását is, ugyanakkor a „valódi" játékok általában néhány alapvető játékelméleti koncepció szinte végtelen számú alkalmazott variánsának egy-egy megjelenése.

A jelen szakdolgozat nem néhány kiragadott „valódi" (pl. sakk) játék programozásával kíván foglalkozni, hanem a játékelméleti alapkoncepciók interaktív és geometriai szemléltetésével. A játékelmélet hasznosságának érzékeltetése céljából a szakdolgozatban elkészített programcsomag egyrészt a közgazdaságtudományban felvetődő, és a játékelmélet (de nem a lineáris programozás) keretei között jól tárgyalható döntés-optimalizációs alapproblémák megértését segíti, másrészt néhány, a pszichológia által felvetett kérdésre esetleg választ adható empirikus kísérletezésre nyújt lehetőséget.

A „Kétszemélyes játékok játékelméleti alapjai" fejezetben röviden összefoglaljuk a szakdolgozathoz szükséges játékelméleti alapkoncepciókat, definíciókat és tételeket. Külön vizsgáljuk a természettel illetve értelmes lénnyel való játékok specifikumait, majd rátérünk a speciális 2x2-es játékok tárgyalására. A fejezetben bemutatunk néhány ismert példát is, amely jobban illusztrálják a 2x2-es játékok fontosságát és érdekességét.

„A kétszemélyes játékok reprezentációi és azok számítógépes megjelenítései" fejezetben a korábban tárgyalt 2x2-es játékok szemléltetésére készített programot mutatjuk be. A program alapvetően két funkciót tölt be. Az egyik a korábban bemutatott speciális játékok interaktívvá tétele, azaz bizonyos feltételekkel lehet „játszani” a felhasználónak. Ez a funkció speciális esetekben alkalmas bizonyos korlátok mellett a bizonytalanságban illetve kockázat mellett hozott döntések „optimizmus-pesszimizmus” dimenziójának pszichológiai vizsgálatához szükséges empirikus adatok gyűjtésére is. Másik funkciója pedig a tárgyalt játékok jobb megértését szolgáló geometriai reprezentációk, szemléltető diagramok vizuális megjelenítése.

A jelen szakdolgozatban található fogalmak, fontosabb példák, tételek illetve állítások megtalálhatóak a döntéselmélet illetve játékelméleti könyvekben, de az egyszerűség kedvéért a szakdolgozat alapvetően követi French (1986) alapművének jelöléseit, fogalmait és formuláit. Fontos kiemelni, hogy számos hazai szerzőtől is vannak kiváló játékelméleti könyvek, mint például Filep (2001) tankönyve, aminek felhasználásával készült pl. a Nash-féle egyensúlypont meghatározását végző programrészlet.

A szerző a kétszemélyes játékok néhány fontos típusára vonatkozó játékelméleti fogalmak, speciális játékok illetve geometriai megjelenítésük rövid összefoglalása mellett (nem törekedve a teljességre) a pszichológiai vizsgálatokban is alkalmazható játékok interaktív programjait és a programok dokumentációt készítette el.

2. A kétszemélyes játékok játékelméleti alapjai

Ebben a fejezetben röviden ismertetjük az általunk vizsgált kétszemélyes játékokra vonatkozó alapismereteket (French, 1986; Osborne and Rubinstein, 1994; Filep, 2001; Kelly, 2003 alapján). A kétszemélyes játékokat érdemes csoportosítani annak alapján, hogy a két játékos mindegyike valódi ember, vagy pedig csak az egyik valódi ember, míg a másik a természet. Az általunk vizsgált esetekben a játék lényege, hogy a játékosok mindegyike kiválaszt egy lehetőséget a számára lehetséges alternatívák közül, figyelembe véve, hogy minden választásának következménye van. A következmény a kontextustól függően lehet nyereség vagy veszteség, és feltételezzük, hogy ezek a következmények kifejezhetőek számokkal (általában valós, de esetenként egész számokkal, vagy esetleg rangszámokkal). Amikor egy valódi ember játszik a természettel, és az emberi játékos számára ismertek a természet alternatíváinak bekövetkezési valószínűségei, akkor kockázat melletti választásról (döntésről) beszélünk. Amennyiben nem ismertek a valószínűségek, akkor bizonytalanság melletti választásról (döntésről) beszélünk. Amikor két valódi ember játszik egymással, akkor stratégiai választásról (döntésről) beszélünk, amelyek lehetnek kooperatívak, kevert motivációjúak, ill. zéró-összegűek.

A főbb változatok közül a dolgozatban a bizonytalanság melletti döntésekkel és a kevert motivációjú stratégiai játékokkal foglalkozunk, és tárgyaljuk a Nash-féle egyensúlypont meghatározását a 2x2-es speciális esetben.

2.1. Játék a természettel

Ebben a fejezetben French (1986), Filep (2001), és Kelly (2003) könyvei alapján a bizonytalanság melletti döntés alapjaival foglalkozunk, azaz amikor a természet állapotainak bekövetkezési valószínűségei ismeretlenek. Ismertetjük a természettel szembeni játék fogalmait, ill. a szakirodalomban elterjedt főbb stratégiai elveit, kritériumait.

A döntési helyzetben két vagy több választási lehetőség adott, de csak egy cselekvés hajtható végre. Ekkor a játékos valamilyen előzetesen meghatározott kritériumnak leginkább megfelelő alternatívát választja, például azt, amelyik a legtöbb hasznot hozza, vagy a legkevesebb veszteséget eredményezi, ill. esetleg több kritérium kombinációját is alkalmazhatja. A döntés természete, hogy a játékos megpróbálja előre megjósolni az általa nem befolyásolható természet állapotának kimenetelét.

Az 1. sz. Táblázatban az S_1, S_2, S_3 (általában az állapotok számát n-el jelöljük) oszlopok a természet állapotait, a D_1, D_2, D_3, D_4 (általában az alternatívák számát m-el jelöljük) sorok pedig a játékos választási lehetőségeit jelölik. A táblázat/mátrix $K_{i,j}$ eleme az *i* választás esetén a természettől függő *j* állapot bekövetkezésekor keletkező kifizetési (nyereség---veszteség) értékeket jelöli.

1. sz. Táblázat: Kifizetési táblázat/mátrix

	S_1	S_2	S_3
D_1	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
D_2	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
D_3	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$
D_4	$K_{4,1}$	$K_{4,2}$	$K_{4,3}$

Egy példát érdemes megfogalmazni, amin keresztül érzékeltethetjük a további fogalmak tartalmi jelentéseit. Legyenek a természet állapotai rendre: S_1=béke, S_2=háború, S_3=zűrzavar/terrorizmus. A választási lehetőségeink pedig rendre: D_1=nemesfém felhalmozás, D_2=ipari befektetés, D_3=üzemanyag felhalmozás, D_4=informatikai beruházások. A kifizetési táblázatban (2. sz. Táblázat) a kifizetési értékek fejezik ki a hipotetikus nyereséget (pozitív értékek) és veszteséget (negatív értékek) valamilyen közös (de pontosan most nem meghatározott) skálán.

2. sz. Táblázat: Kifizetési táblázat/mátrix befektetési alternatívák és társadalmi helyzet esetén

	S_1=béke	S_2=háború	S_3=zűrzavar/terrorizmus
D_1=nemesfém felhalmozás	1	5	2
D_2=ipari befektetés	5	-2	-1
D_3=üzemanyag felhalmozás	3	8	-4
D_4=informatikai beruházások	6	3	2

A játékos előre ismeri a kifizetési táblázat adatait, azaz ismeri, hogy a természet különböző bekövetkezési állapotai esetében mekkora kifizetéshez juthat. De azt nem tudja előre, hogy mi fog ezek közül bekövetkezni, ezért bizonytalanság mellett kell meghoznia a döntését, azaz választania kell, hogy mit tegyen. A példában, ha háború lesz, akkor a legnagyobb nyereséget (8 egységet) akkor érhetjük el, ha sok üzemanyagot halmozunk fel, de zűrzavar esetén a felhalmozott üzemanyagot nem tudjuk megbízhatóan értékesíteni, ezért végül sokat veszíthetünk (4 egységet a példában).

A továbbiakban néhány hasznos és ismert elvet mutatunk be, amelyek különböző motiváltságokat fejeznek ki, és amelyek alkalmazásai (nem feltétlenül) különböző nyereségeket illetve veszteségeket eredményezhetnek még akár azonos természeti állapot esetében is.

A döntés alapjául szolgáló elv/szabály többféle lehet, és feltételezzük, hogy minden egyes D_i (ahol i = 1,…,m) választási lehetőséghez létezik egy kimeneti érték, amit V_i-vel jelölünk. A lehetséges alternatívákat általánosan S_j-vel jelöljük (ahol j = 1,…,n). Az *i* választás esetén a természettől függő *j* állapot bekövetkezésekor keletkező kifizetési (nyereség---veszteség) értékeket $K_{i,j}$ jelöli.

További gyakran használt jelöléseink:

$$\text{Kmax}_i = \overset{n}{\underset{j=1}{Max}} K_{i,j}$$

$$\text{Kmin}_i = \overset{n}{\underset{j=1}{Min}} K_{i,j}$$

2.1.1. Minimax kritérium (Neumann and Morgenstern, 1944)

A minimax elv esetében amennyiben a kifizetési értékek nyereséget jelentenek, akkor a minimax kritériumot választókra jellemző, hogy egyrészt törekednek a maximális nyereségre, ugyanakkor biztonságosan kívánják ezt elérni, ezért azt a lehetőséget választják, amelyik legalább egy minimális nyereséget eredményez a lehető maximumok közül.

Ha a kifizetési értékek veszteséget jelentenek, akkor a minimax kritériumot választókat jellemezhetjük úgy, hogy megpróbálják elkerülni a legnagyobb veszteséget, azaz választásukkal minimalizálják annak lehetséges értékét.

$$V_i = \text{Kmax}_i$$

Ekkor a választás a k-dik alternatíva, azaz

$$D_k, \text{ ha } V_k = \overset{m}{\underset{i=1}{Min}} V_i$$

A 2. sz. Táblázat alapján a választható alternatívák maximális értékei:

$D_1 \rightarrow V_1 = 5$ $D_2 \rightarrow V_2 = 5$ $D_3 \rightarrow V_3 = 8$ $D_4 \rightarrow V_4 = 6$

Így két minimum választás lehetséges, a D_1 és a D_2 .

2.1.2. Maximin kritérium (Wald, 1950)

A választás értelmezhető úgy, hogy a játékos a lehetőségek minimális nyereségét próbálja maximalizálni. Minden egyes lehetőség esetében előbb meghatározza a természettől függő minimális nyereséget, mint egyfajta biztonsági értéket, majd azt a lehetőséget választja, amelyik a biztonság mellett a legnagyobb nyereséget biztosíthatja számára. A minimális nyereség maximalizálása egyfajta pesszimizmusra utal.

$$V_i = \text{Kmin}_i$$

Ekkor a választás a k-dik alternatíva, azaz

$$D_k, \text{ ha } V_k = \underset{i=1}{\overset{m}{Max}} V_i \quad .$$

A 2. sz. Táblázat alapján a választható alternatívák minimális értékei:

$D_1 \rightarrow V_1 = 1$ $\quad$ $D_2 \rightarrow V_2 = -2$ $\quad$ $D_3 \rightarrow V_3 = -4$ $\quad$ $D_4 \rightarrow V_4 = 2$

Így egy maximum választás lehetséges, a D_4 .

2.1.3. Minimax megbánás kritérium (Savage, 1951)

A minimax megbánás azt jelenti, hogy nyereséget jelentő kifizetést feltételezve, a maximális nyereség és az aktuális nyereség közti különbséget a játékos veszteségnek fogja fel, ezért a megbánás értékeire a minimax elvet használja, azaz a megbánás maximumát próbálja minimalizálni, azaz nem akar olyan döntést hozni, amit később megbánna.

$$\text{megbánás} = r_{i,j} = \underset{i=1}{\overset{m}{Max}}[K_{i,j}] - K_{i,j}$$

$$V_i = \underset{j=1}{\overset{n}{Max}}\, r_{i,j}$$

Ekkor a választás a k-dik alternatíva, azaz

$$D_k, \text{ ha } V_k = \underset{i=1}{\overset{m}{Min}} V_i \quad .$$

A 2. sz. Táblázat alapján a természet állapotaihoz tartozó maximális értékek:

$S_1 \rightarrow 6$ $\quad$ $S_2 \rightarrow 8$ $\quad$ $S_3 \rightarrow 2$

Ebből kiszámíthatóak a megbánás értékei, azaz az r_{ij} értékek, amelyek után a minimax elvet alkalmazza a játékos. A 3. sz. Táblázatban megadjuk a megbánás értékeket.

3. sz. Táblázat: Befektetési alternatívák és társadalmi helyzet esetén a megbánás értékek

	S_1=béke	S_2=háború	S_3=zűrzavar/terrorizmus
D_1=nemesfém felhalmozás	5	3	0
D_2=ipari befektetés	1	10	3
D_3=üzemanyag felhalmozás	3	0	6
D_4=informatikai beruházások	0	5	0

A 3. sz. Táblázat alapján a választható alternatívák maximális értékei:

$D_1 \to V_1 = 5$ $D_2 \to V_2 = 10$ $D_3 \to V_3 = 6$ $D_4 \to V_4 = 5$

Így a minimax értelmében ezek közül két minimum van, azaz két választás lehetséges, a D_1 és a D_4 .

2.1.4. Maximax kritérium

A maximax elvet képviselők az optimisták, azt gondolják, hogy a legnagyobb nyerséget tudják realizálni, ezért csak azt az alternatívát keresik, amelyik ezt biztosítja.

$$V_i = K\max_i$$

Ekkor a választás a k-dik alternatíva, azaz

$$D_k, \text{ ha } V_k = \underset{i=1}{\overset{m}{Max}} V_i$$

A 2. sz. Táblázat alapján a választható alternatívák maximális értékei:

$D_1 \to V_1 = 5$ $D_2 \to V_2 = 5$ $D_3 \to V_3 = 8$ $D_4 \to V_4 = 6$

Így egy maximum választás lehetséges, a D_3 .

2.1.5. Nem elégséges ok kritérium (Laplace, 1825)

Ha a játékos nem tud semmilyen „okot” találni a természet állapotának becslésére, akkor választhatja az átlagos nyereséget, mint elvet, és ekkor a legnagyobb átlagos nyereséget fogja választani.

$$V_i = \frac{1}{n} \sum_{j=1}^{n} K_{i,j}$$

Ekkor a választás a k-dik alternatíva, azaz

$$D_k, \text{ ha } V_k = \underset{i=1}{\overset{m}{Max}} V_i \quad .$$

A 2. sz. Táblázat alapján a választható alternatívák átlagos értékei:

$D_1 \to V_1 = 8/3$ $D_2 \to V_2 = 2/3$ $D_3 \to V_3 = 7/3$ $D_4 \to V_4 = 11/3$

Így egy maximum választás lehetséges, a D_4 .

2.1.6. Optimizmus-pesszimizmus index (Hurwicz, 1951)

A két szélsőséges stratégia, a pesszimista (maximin) és az optimista (maximax) a gyakorlatban nehezen elfogadható, az emberek elképzelései általában a két szélsőség közé esnek. Az egyénre jellemző lehet egy optimista-pesszimista index érték, ami úgy származtatható, hogy a játékos egy magára jellemző (α-val jelölt) „pesszimizmus fok”-kal súlyozza a pesszimista stratégiát, illetve az (1-α)-val értelmezett „optimizmus fok”-kal pedig az optimista stratégiát súlyozza és a súlyozott összeg lesz az optimista-pesszimista index. A játékos választását adott α mellett ezen értékek maximuma eredményezi, azaz:

$$\mathrm{V_i} = \alpha \mathrm{Kmin}_i + (1-\alpha)\mathrm{Kmax}_i$$

Ekkor a választás a k-dik alternatíva, azaz

$$\mathrm{D_k}, \text{ ha } \mathrm{V_k} = \operatorname*{Max}_{i=1}^{m} V_i \quad .$$

A 4. sz. Táblázatban felhasználva a 2. sz.Táblázat adatait, különböző lehetséges α pesszimizmus fok mellett meghatározzuk a legjobb döntést és a kapcsolódó kifizetési értéket. A táblázatban szövegesen fogalmaztuk meg a döntést, és jól látható, hogy a saját döntésében optimistán bízó játékos a példában az üzemanyag felhalmozását választja, és ezáltal 8 egységnyi nyerségre próbál szert tenni. Ahogy csökken egy játékosnak a saját döntésébe való hite/optimizmusa, úgy nő a pesszimizmusa, és a pesszimista véglet az informatikai beruházások választásához vezet.

4. sz. Táblázat. A pesszimizmus fokától függő választások

ha α=0	akkor a legjobb választás	D_3 (üzemanyag felhalmozás)	aminek értéke = 8
ha α=0.1	akkor a legjobb választás	D_3 (üzemanyag felhalmozás)	aminek értéke = 6.8
ha α=0.2	akkor a legjobb választás	D_3 (üzemanyag felhalmozás)	aminek értéke = 5.6
ha α=0.3	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 4.8
ha α=0.4	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 4.4
ha α=0.5	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 4.0
ha α=0.6	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 3.6
ha α=0.7	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 3.2
ha α=0.8	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 2.8

ha α=0.9	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 2.4
ha α=1	akkor a legjobb választás	D_4 (informatikai beruházások)	aminek értéke = 2

2.1.7. Az optimizmus-pesszimizmus index kísérletes becslése

Ha egy személy optimizmus-pesszimizmus értéke ismert, akkor a Hurwicz-elv alapján viszonylag egyszerűen tehetünk becslést arra, hogy különböző döntési helyzetekben (a természettel való játékhelyzetben természetesen) a játékos hogyan fog választani. Az azonnal megfogalmazódó kérdés, hogy honnan tudhatjuk a játékos optimizmus-pesszimizmus értékét. Ezt az értéket természetesen nem tudhatjuk, de alkalmas „játék-kísérletekkel” van esélyünk becsülni.

Vizsgáljunk egy olyan játékot, aminek kifizetés mátrixa a következő (5. sz. Táblázat).

5. sz. Táblázat: Optimizmus-pesszimizmus kísérleti kifizetési táblázat/mátrix

	S_1	S_2
D_1	1	0
D_2	x	x

Ekkor a D_1 választás minimális kifizetési értéke 0, maximális értéke pedig 1. A D_2 választás esetében a minimum és a maximum egyenlőek, és értékük x. Ebből következik, hogy a Hurwicz-elv alapján a D_1 választás értéke az α optimizmus-pesszimizmus (egész pontosan a jelenlegi jelölésekkel ez a pesszimizmust fejezi ki) mellett egyenlő 1-α-val, míg a D_2 választás értéke egyenlő x-szel. A 6. sz. Táblázatban összefoglaltuk az érvelés lényegét.

6. sz. Táblázat: Optimizmus-pesszimizmus kísérleti értékei a Hurwicz-elv alapján

	S_1	S_2	min	max	max * (1-α) + min * α
D_1	1	0	0	1	1 - α
D_2	x	x	x	x	x

Ebből következik, hogy ha a játékos indifferens a D_1 és a D_2 választásában, akkor

$$1 - \alpha = x, \qquad \text{azaz} \qquad \alpha = 1 - x.$$

A kísérletes meghatározás során az 5. sz. Táblázatnak megfelelően az x értékeit 0 és 1 között változtatjuk (pl. 0.1-ként), és a felkínált kifizetési értékeket figyelembe véve a játékos minden egyes esetben választ az alternatívák közül.
A kísérletben három válaszlehetőség közül választhat a játékos:

- D_1-et választja,
- indifferens a D_1 és D_2 közötti választásban,
- D_2-t választja.

Amikor a játékos a saját választásában elbizonytalanodik, azaz a játékos indifferens választást ad a D_1 és a D_2 közötti választásában, akkor az aktuális x érték lesz a játékos α értéke.

A kísérlet során előfordulhat, hogy

- a játékos nem mond indifferens választást, de vált az alternatívák között. Ekkor az egyszerűség kedvéért, a váltás (a váltást az „x" növekvő sorrendje mellett értjük) előtti és a váltás utáni „x" értékek átlagát vesszük („xátl" -al jelöljük) és az „1 - xátl" értéket tekintjük a játékos α értékének.
- a játékos többször is „szomszédos" indifferens választást ad (azaz a növekvő x értékek mellett az indifferens válaszok között nem történik váltás). Ekkor megint az egyszerűség kedvéért, az indifferens választásokhoz tartozó „x" értékek átlagát vesszük és az „1 - xátl" értéket tekintjük a játékos α értékének.
- a játékos többször és nem következetesen ad indifferens választást, azaz az indifferens és a határozott választások egymást váltják. Ekkor csak annyit mondhatunk, hogy a játékos nem elég következetes ahhoz, hogy ilyen módon meghatározzuk az α értékét. (A program az adatbázisba ekkor is rögzíti a választásokat a további adatelemzés céljából.)

2.2. Játék értelmes ellenféllel

Ebben a fejezetben Osborne and Rubinstein (1994), Filep (2001), Canty (2003), és Kelly (2003) könyvei alapján azt az esetet mutatjuk be, amikor két értelmes játékos egymás ellen játszik. A szakirodalomban található főbb változatok közül a dolgozatban a kevert motivációjú ún. 2x2-es stratégiai játékokkal foglalkozunk, és tárgyaljuk a Nash-féle egyensúlypont (Nash, 1951; Canty, 2003) meghatározását is a 2x2-es speciális esetben. Az egyensúlypont meghatározását a dolgozathoz kapcsolódó interaktív program a Filep (2001) könyvében (8. fejezet) levő algoritmus alapján végzi.

2.2.1. A 2x2-es játékok alapfogalmai, jelölései

Amikor két értelmes játékos játszik egymás ellen, akkor mind a két játékosnak külön-külön adott a játékosok választásaitól függő kifizetéseinek a mátrixa, amit szokás a 7. sz. Táblázatnak megfelelően egy közös táblázatban megadni. A 7. sz. Táblázatban a speciális 2x2-es változatra adjuk meg a jelöléseket, de nyilvánvaló az általánosítás lehetősége több alternatív válaszlehetőség esetére.

7. sz. Táblázat: a 2x2-es játékok kifizetés mátrixa

	C_1	C_2
R_1	$r_{1,1}$; $c_{1,1}$	r_{12} ; $c_{1,2}$
R_2	$r_{2,1}$; $c_{2,1}$	$r_{2,2}$; $c_{2,2}$

A sorjátékost R-el (aki az R_1 és az R_2 alternatívák/stratégiák közül választhat), az oszlopjátékost C-vel jelöljük (aki az C_1 és az C_2 alternatívák/stratégiák közül választhat), a cellákban (az indexek egy mátrix elemeinek a szokásos módon történő indexelését követik) pedig a kifizetési értékek szerepelnek. Igazodva a játékosok és az általuk választott stratégia elnevezéséhez (például $r_{2,1}$ az R_2 és a C_1 választások mellett a sorjátékos, míg $c_{2,1}$ az oszlopjátékos kifizetési értékét jelöli). A felhasználói program és az egyszerűség kedvéért az R játékost 1-es játékosnak, a C játékost pedig 2-es játékosnak, a stratégiákat pedig 1-es illetve 2-es sorszámú stratégiáknak is nevezzük

A játék abból áll, hogy mind a két játékos választ egy-egy alternatívát, majd a kifizetési mátrix alapján meghatározzák, mindkét játékos esetében a kifizetés értékét. A 8. sz. Táblázatban a jól ismert fogoly-dilemma (Zagare, 1984) egy változatát adjuk meg, amelyből leolvasható, hogy ha a sorjátékos tagad, de az oszlopjátékos vall, akkor a sorjátékos kifizetési értéke -10 (azaz például 10 év börtön), az oszlopjátékosé pedig 0 (azaz például szabadon engedik). A fogoly-dilemma játékban a tagadást szokás kooperatív, a bevallást pedig versengő választásnak nevezni. Ekkor azt láthatjuk, hogy egy kooperatív játékos (azaz aki a kooperatív alternatívát választja) egy versengővel szemben nagy veszteségeket érhet el, ugyanakkor két versengő játékos együtt is komoly veszteségeket okoznak egymásnak.

8. sz. Táblázat: a fogoly-dilemma egy változata

		C	
		tagad	bevall
R	tagad	-1 ; -1	-10 ; 0
	bevall	0 ; -10	-5 ; -5

Érdemes megemlíteni, hogy a fogoly-dilemma kifizetési mátrixát a szakirodalom nem mindig a 8. sz. Táblázatnak megfelelően írja fel, pl. Filep (2001) könyvében a tagadás és bevallás sorrendje mind a sorjátékos, mind az oszlopjátékos esetében fel van cserélve. Ez nyilván a játék lényegét nem érinti, de a további félreértések elkerülése végett külön érdemes odafigyelni, hogy a számítások eredményét a kifizetési mátrix és a kapcsolódó tartalommal együtt kell kezelni.

2.2.2. A 2x2-es játékok Nash-féle egyensúlypontja

A játékelmélet egyik fontos alapfogalma a játék egyensúlypontja, aminek most a Nash (1951)-féle változatának 2x2-es formáját vizsgáljuk. A fejezetben a kapcsolódó fogalmak és alapvető tételek ismertetésekor elsősorban Canty (2003) ill. Filep (2001) könyveire támaszkodunk, a bizonyítások Nash (1951) tanulmányában megtalálhatóak. A kapcsolódó program Filep (2001) könyvének 8. fejetében megadott számítási módokat alkalmazza.

Mielőtt rátérnénk a formális definíciókra, áttekintjük a kétszemélyes játékok néhány alapvető, és a Nash (1951)-féle egyensúlyponthoz kapcsolódó kérdésfeltevését és fogalmát. Egy kétszemélyes játék esetében alapvető kérdés, hogy a játékosok milyen elvet kövessenek stratégiáik választásakor, feltételezve, hogy a játékosok a kifizetési értékeiket tekintve rendelkeznek valamilyen preferencia-sorrenddel, amit következetesen érvényesíteni akarnak. A játék kifizetési mátrixa alapján a játékosok választanak a két alternatív lehetőségük közül egy-egy stratégiát (ezt a stratégia párt nevezzük a játék kimenetelének), és a kifizetési mátrixnak a kimenetel megfelelő cellájában szereplő értékek alapján kapják meg a nyereségüket/veszteségüket. Az általunk vizsgált 2x2-es esetben a játékosok mindegyike két lehetőség közül választhat, és mindig azt kell mérlegelnie, hogy vajon a másik játékos mit választ, hiszen annak ismeretében a döntése könnyű lenne. Ekkor egyszerűen azt az alternatívát érdemes választania, ami a másik feltételezett választásának oszlopában (vagy sorában) a számára legkedvezőbb válasz, azaz az adott körülmények mellett a legnagyobb nyereséget biztosítja. Az egyensúlypont egy olyan helyzet, amelytől egyik játékosnak sincs érdeke eltérni, azaz mindkét játékos a számára legkedvezőbb stratégiát választja, feltéve, hogy a másik játékos sem tér el az elvileg számára legkedvezőbb stratégia választásától. Az ilyen egyensúlyponthoz kapcsolódó stratégiákat tiszta egyensúlyi stratégiáknak, illetve az egyensúlypontot tiszta egyensúlypontnak nevezzük (Filep,2001).

Előfordulhat, hogy a játékhoz nincs tiszta egyensúlypont, ilyenkor célszerű kiterjeszteni a választási lehetőségeinket a lehetséges választási lehetőségekre vonatkozó valószínűségi eloszlásokra. Ekkor a kifizetési értékeket az ún. várható kifizetési értékek váltják fel, és a legkedvezőbb választ is (hasonlóan a tiszta egyensúlypontok esetéhez) a várható kifizetési értékek alkalmazásával érhetjük el.

1. Definíció: Egy, a 7. sz. Táblázatnak megfelelő 2x2-es játék esetén a sorjátékos R_m stratégiája a legjobb válasz az oszlopjátékos C_n stratégiájára, ha

$$r_{i,n} \leq r_{m,n} \qquad (i = 1,2).$$

Hasonlóan definiáljuk a másik játékos legjobb válaszát, azaz az oszlopjátékos C_n stratégiája a legjobb válasz a sorjátékos R_m stratégiájára, ha

$$c_{m,j} \leq c_{m,n} \qquad (j = 1,2).$$

Az (R_m , C_n) stratégiapárt Nash-féle tiszta egyensúlypontnak nevezzük, ha R_m és C_n legjobb válaszok a másik játékos stratégiájára nézve, azaz R_m a legjobb válasz C_n-re, és C_n a legjobb válasz R_m-re.

2. Definíció: Egy, a 7. sz. Táblázatnak megfelelő 2x2-es játék esetén:

- a sorjátékos kevert stratégiáján a

$$P = (p, 1-p) \qquad \text{ahol} \quad 0 \le p \le 1$$

valószínűségi eloszlást értjük, ahol p az R_1 stratégiához, (1-p) pedig az R_2 stratégiához tartozó valószínűséget jelöli.

- az oszlopjátékos kevert stratégiáján a

$$Q = (q , 1-q) \qquad \text{ahol} \quad 0 \le q \le 1$$

valószínűségi eloszlást értjük, ahol q a C_1 stratégiához, (1-q) pedig a C_2 stratégiához tartozó valószínűséget jelöli.

- a sorjátékos várható kifizetési értéke P és Q kevert stratégiák esetén:

$$E_R(P , Q) = p [q\, r_{1,1} + (1 - q)\, r_{1,2}] + (1 - p) [q\, r_{2,1} + (1 - q)\, r_{2,2}]$$

- az oszlopjátékos várható kifizetési értéke:

$$E_C(P , Q) = p [q\, c_{1,1} + (1 - q)\, c_{1,2}] + (1 - p) [q\, c_{2,1} + (1 - q)\, c_{2,2}]$$

3. Definíció: Egy, a 7. sz. Táblázatnak megfelelő 2x2-es játék esetén a sorjátékos P^* stratégiája a legjobb kevert válasz az oszlopjátékos Q kevert stratégiájára, ha

$$E_R(P , Q) \le E_R(P^* , Q) \qquad \text{(a sorjátékos minden P kevert stratégiája esetén).}$$

Hasonlóan definiáljuk a másik játékos legjobb kevert válaszát, azaz az oszlopjátékos Q^* kevert stratégiája a legjobb válasz a sorjátékos P kevert stratégiájára, ha

$$E_C(P , Q) \le E_C(P , Q^*) \qquad \text{(az oszlopjátékos minden Q kevert stratégiája esetén).}$$

Az (P^* , Q^*) kevert stratégia párt **Nash-féle kevert egyensúlypontnak** nevezzük, ha P^* és Q^* legjobb kevert válaszok a másik játékos kevert stratégiájára nézve, azaz P^* a legjobb kevert válasz Q^*-ra, és Q^* a legjobb válasz P^*-ra:

- $E_R(P , Q^*) \le E_R(P^* , Q^*)$ (a sorjátékos minden P kevert stratégiája esetén)
- $E_C(P^* , Q) \le E_C(P^* , Q^*)$ (az oszlopjátékos minden Q kevert stratégiája esetén)

1. Tétel (Nash, 1951): Egy, a 7. sz. Táblázatnak megfelelő 2x2-es játék esetén mindig van Nash-féle kevert egyensúlypont.

A 2x2-es játékok esetében a Nash-féle kevert egyensúlypontok meghatározását az algoritmikus megoldás mellett érdemes grafikusan is keresni (pl. Canty, 2003; Filep, 2001). Erre jó alapot ad a legjobb kevert válaszok ábrázolása egy koordináta-rendszerben, ahol pl. a vízszintes tengelyen 0-tól 1-ig a sorjátékos (p, 1 - p) kevert stratégiájának p értékét változtatva, ábrázolhatjuk az oszlopjátékos legjobb válaszainak q értékeit, és fordítva. Ekkor a két ábrát egyben ábrázolva, a metszéspontok lesznek a Nash-féle kevert egyensúlypontok. Amennyiben a p-re és/vagy a q-ra a 0 és/vagy 1 adódik, akkor nyilvánvaló, hogy ezek a kevert stratégiák tulajdonképpen megfelelnek a tiszta stratégiákkal: (0, 0), (1, 0), (0, 1), (1, 1).

A 2x2-es játékok Nash-féle egyensúlypontjának megkeresése és geometriai ábrázolása

Ebben a fejezetben leírjuk - de nem vezetjük le-, hogy hogyan találjuk meg a Nash-féle egyensúlypontokat és azokat miként kell ábrázolnunk. (Filep, 2001)

Legyenek I. és II. játékosok fizetési mátrixai:

A

$a_{1,1}$	$a_{1,2}$
$a_{2,1}$	$a_{2,2}$

B

$b_{1,1}$	$b_{1,2}$
$b_{2,1}$	$b_{2,2}$

Valamint legyenek p(p, 1-p), $0 \leq p \leq 1$ és q(q,1-q), $0 \leq q \leq 1$ a játékosok stratégiái. A p és q stratégiák mértani helyét a $0 \leq p \leq 1$, $0 \leq q \leq 1$ oldalú egységnégyzet pontjai adják. Vezessük be a következő jelöléseket:

- $a_{1,1}-a_{1,2}-a_{2,1}+a_{2,2}=\alpha$
- $a_{2,2}-a_{1,2}=a$
- $b_{1,1}-b_{1,2}-b_{2,1}+b_{2,2}= \beta$
- $b_{2,2}-b_{2,1}=b$

Ezek után:

(1) $\alpha(1-p^0)q^0-a(1-p^0)\leq 0$

(2) $\alpha p^0q^0-ap^0\geq 0$

Az I. játékos egyensúlypontjait ezen egyenlőtlenség-rendszernek a [0,1]×[0,1] egységnégyzetbe eső megoldásai adják.

A megoldás szerkezete α, β és a, b értékétől függ, ahol $\lambda=a/\alpha$ és $\mu=b/\beta$:
A megoldások λ, α és a függvényében:

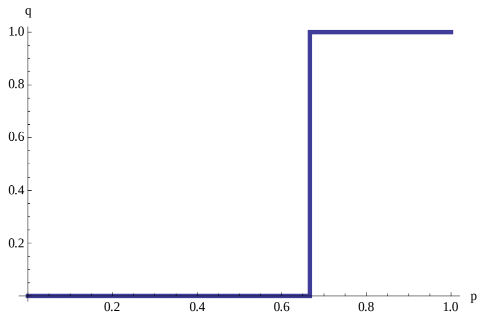
1. $\alpha=0$ és $a=0$, akkor a teljes négyzet adja a megoldást.
2. $\alpha=0$ és $a>0$, akkor (0,0) és (0,1) pont közötti egyenes
3. $\alpha=0$ és $a<0$, akkor (1,0) és (1,1)
4. $\alpha!=0$ és $\lambda=0$, akkor (0,0) és (1,0) és (1,1) pontok összekötésével kapjuk a megoldást
5. $\alpha>0$ és $\lambda>0$ és $\lambda<1$, akkor (0,0) és $(0, \lambda)$ és $(1, \lambda)$ és (1,1) pontok összekötésével
6. $\alpha<0$ és $\lambda>0$ és $\lambda<1$, akkor (1,0) és $(1, \lambda)$ és $(0, \lambda)$ és (0,1) pontok összekötésével
7. $\alpha<0$ és $\lambda<0$, akkor (0,0) és (0,1) pont közötti egyenes
8. $\alpha>0$ és $\lambda<0$, akkor (1,0) és (1,1) pont közötti egyenes
9. $\alpha<0$ és $\lambda=0$, akkor (0,1) és (0,0) és (1,0) pontok összekötésével
10. $\alpha>0$ és $\lambda=1$, akkor (0,0) és (0,1) és (1,1) pontok összekötésével
11. $\alpha<0$ és $\lambda=1$, akkor (0,1) és (1,1) és (1,0) pontok összekötésével
12. $\alpha>0$ és $\lambda>1$, akkor (0,0) és (0,1) pont közötti egyenes
13. $\alpha<0$ és $\lambda>1$, akkor (1,0) és (1,1) pont közötti egyenes

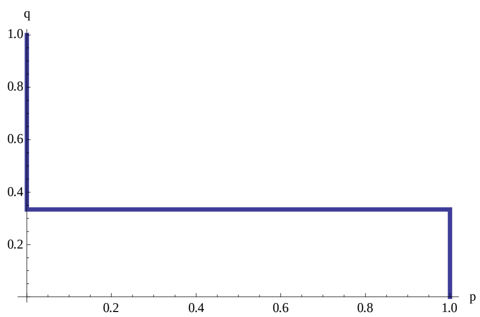
μ, β és b esetében a megoldások hasonlóan alakulnak, sok esetben szimmetrikusan:

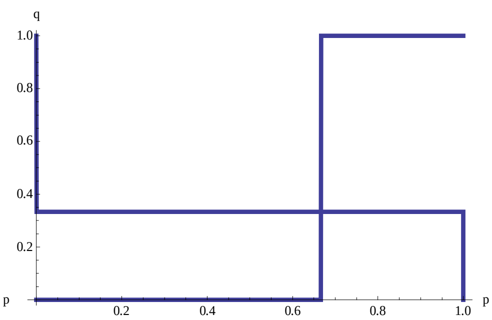
1. $\beta=0$ és $b=0$, akkor a teljes négyzet adja a megoldást.
2. $\beta=0$ és $b>0$, akkor (0,0) és (1,0) pont közötti egyenes
3. $\beta=0$ és $b<0$, akkor (0,1) és (1,1)
4. $\beta!=0$ és $\mu=0$, akkor (0,0) és (0,1) és (1,1) pontok összekötésével kapjuk a megoldást
5. $\beta>0$ és $\mu>0$ és $\mu<1$, akkor (0,0) és $(\mu,0)$ és $(\mu,1)$ és(1,1) pontok összekötésével
6. $\beta<0$ és $\mu>0$ és $\mu<1$, akkor (1,0) és $(\mu,0)$ és $(\mu,1)$ és(0,1) pontok összekötésével
7. $\beta<0$ és $\mu<0$, akkor (0,0) és (1,0) pont közötti egyenes
8. $\beta>0$ és $\mu<0$, akkor (0,1) és (1,1) pont közötti egyenes
9. $\beta<0$ és $\mu=0$, akkor (0,1) és (0,0) és (1,0) pontok összekötésével

10. β >0 és μ=1, akkor (0,0) és (1,0) és (1,1) pontok összekötésével
11. β <0 és μ=1, akkor (1,0) és (1,1) és (0,1) pontok összekötésével
12. β >0 és μ>1, akkor (0,0) és (1,0) pont közötti egyenes
13. β <0 és μ>1, akkor (0,1) és (1,1) pont közötti egyenes

Megjegyzés: Az esetekre bontást bizonyos helyeken akár össze is vonhatnánk, ám a könnyebb érthetőség miatt a lehető legrészletesebben került leírásra. Illetve egyes feltételek λ, μ, α, β és a, b függvényében másképp is megfogalmazhatóak, úgy hogy az eredmény ekvivalens maradjon.







3. A kétszemélyes játékok reprezentációi és azok számítógépes megjelenítései

Ebben a fejezetben bemutatjuk, hogy az előbb ismertetett játékokat milyen formában jelenítjük meg számítógépen, milyen programozási eszközöket használtunk. A főbb szempontok a következők lesznek:

- A programozási háttér bemutatása
  - Részletesen kitérünk a Java Applet technológiára
- A program szerkezete és működése
- Megismerkedünk a felhasználói felülettel

3.1. A programozási háttér bemutatása

Ebben a fejezetben bemutatjuk, hogy a program milyen eszközökkel készült. Kitérünk az alkalmazott technológiák előnyeinek és hátrányainak ismertetésére. Megadjuk, hogy a program működtetéséhez milyen egyéb műveleteket kellett elvégeznünk ahhoz, hogy helyesen működjenek.

3.1.1. A Java Applet

Az Applet egy olyan Java nyelven írt program, ami beágyazható HTML oldalakba, akár egy kép. Amikor Java technológiát engedélyező böngészőt használunk ahhoz, hogy megnézzünk egy ilyen beágyazott Appletet, az Applet kódja letöltődik a rendszerünkbe és a böngésző JVM-e végrehatja.
(http://java.sun.com/applets/)

Egy Appletnek mindenképp a java.applet.Applet osztály alosztályának kell lennie. Az Applet osztály szolgáltat egy sztenderd interfészt az Applet és a böngésző környezete között. A Swing szolgáltat egy speciális alosztályt az Applethez, javax.swing.JApplet. A JAppletet kell használnunk minden olyan Applethez, ami használ Swing komponenseket GUI-k létrehozásához.
(http://java.sun.com/docs/books/tutorial/deployment/applet/index.html)

3.1.2. Applet: a fordítástól a böngészőben való megjelenítésig

Ahhoz, hogy böngészőben meg tudjunk jeleníteni egy Appletet először le kell fordítanunk a forráskódot, majd egy JAR fájlba kell csomagolnunk. Ha Eclipse integrált fejlesztői környezetet használunk, akkor ezt a legegyszerűbben így tudjuk megtenni:

- Fájl/Export
- Java/JAR file
- Next
- Kiválasztjuk a forrást, amit exportálni akarunk
- A „Select the export destination:” alatti mezőbe kitöltjük, hogy hova mentsük a fájlt
- Finish

Ezen az egyszerű megoldáson túl még több beállítást is alkalmazhatunk, de most ezeket nem tárgyaljuk.

Egy Appletet kétféleképpen tudunk elindítani:

1. Megtehetjük, hogy specifikáljuk az Applet indítási beállításait az <applet> tagen belül. Ez egy régi módja egy Applet indításának, és komoly biztonsági korlátozásokat ró az Appletre.
2. A másik mód, hogy Java Network Launch Protocolt használunk (JNLP).

(http://java.sun.com/docs/books/tutorial/deployment/applet/deployingApplet.html)

A JNLP fájlok struktúrája

Gyakran használt elemek és attribútumok egy JNLP fájlban. A JNLP kódolását az XML fájl elején kell megadni. Pl.: <?xml version="1.0" encoding="UTF-8"?>

Elemek	Attribútumok	Leírás
jnlp		A gyökér XML elem a JNLP fájlban
	spec	Lehet: 1.0,1.5,6.0 vagy 1.0+ A minimum JNLP specifikációt mutatja, amivel a JNLP fájl tud dolgozni.
	codebase	Az alap elhelyezkedés, amihez a relatív útvonalakat mérni kell.
	href	A JNLP fájl URL-e.
information		jnlp gyermeke
title		information gyermeke, a RIA címe
vendor		information gyermeke, a RIA ellátója
resources		jnlp gyermeke, leír minden forrást amire a RIA-nak szüksége van.
j2se		resources gyermeke, Java verzió
	version	verzió megadása
	href	az URL
jar		resources gyermeke, A jar fájl ami része a RIA classpathának
	href	az URL
	main	true/false, tartalmazza-e a jar fájl a main metódust
applet-desc		jnlp gyermeke, azt mutatja, hogy a JNLP fájl egy applethez készült
	name	az Applet neve
	main-class	a main Applet osztály neve
	width	Applet szélessége px-ben
	height	Applet magassága pixelben
param		paraméterek, amit az Appletnek küldünk
	name	paraméter neve
	value	paraméter értéke
update		jnlp gyermeke, a RIA miképpen hajtsa végre a frissítéseket
	check	lehet always, timeout, vagy background

http://java.sun.com/docs/books/tutorial/deployment/deploymentInDepth/jnlpFileSyntax.html

A HTM fájl

A HTM fájl felépítése a következőképpen alakul. Csak az Appletre vonatkozó részt emeljük ki.

```
<script src="http://www.java.com/js/deployJava.js" type="text/javascript">
</script>
<script type="text/javascript">
	var attributes = { code:'keret.JatekApplet',  width:800, height:600} ;
	var parameters = {jnlp_href: 'szakdolgozat.jnlp'} ;
	deployJava.runApplet(attributes, parameters, '1.6');
</script>
```

A deployJava.js tartalmazza az Applet elindításához szükséges runApplet(attributes, parameters, minimumVersion) függvényt. A <script> tagban definiáljuk az attribútumokat. Itt adjuk meg azt az osztályunkat, ami alosztálya az Applet vagy JApplet osztálynak (csomagnév.Osztálynév formában). További attribútumként a szélességet és a magasságot is beállíthatjuk.

A paraméterünk nem más, mint a JNLP fájl neve, elérési úttal. Ebben az esetben megállapítható, hogy HTM fájl és a JNLP fájl egy mappában található. Ezek után nincs más dolgunk, mint a runApplet függvény meghívása.

Ha mindezzel elkészültünk az Appletünket már meg is tekinthetjük a böngészőben.

3.1.3. Tanúsítvány probléma

Amikor azt szeretnénk, hogy az általunk készített Applet hozzáférhessen a felhasználó állományaihoz és különböző fájlműveleteket, illetve egyéb biztonsági kockázattal járó műveleteket hajtson végre, akkor az elkészített JAR fájlt alá kell írnunk. Így, amikor a felhasználó betölti az oldalt, először egy figyelmeztő ablakkal találja magát szembe, hogy megbízik-e az Appletben, illetve annak készítőjében.

Megjegyzés: *„… nem hiteles forrásból származó, u.n. self-signed, azaz önkezűleg aláírt SSL tanúsítványok a legtöbb felhasználóban nem keltenek elég bizalmat, az elterjedt böngészőprogramok kiemelten figyelmeztetik a felhasználókat, hogy az SSL tanúsítvány érvénytelen, a kibocsátója nem ismert, nem megbízható. A VeriSign®, a Thawte® és a RapidSSL® mind elismert CA (Certificate Authority), root certificate-jüket az elterjedt* *böngészőprogramok beépítve tartalmazzák évek óta, így az általuk hitelesített tanúsítványokat automatikusan felismerik és elfogadják.” (http://www.ezit.hu/?cat=158)*

Ahhoz, hogy egy megfelelő kulcsot magunk létrehozzunk, a következő lépéseket kell tennünk.

A Java keytool eszközét használjuk. Parancssorban a következő lépéseket kell végrehajtanunk:

1. keytool -genkey -alias *alias* -keystore *.keystore* , ahol az alias a kulcs neve, a keystore pedig hogy hol tároljuk a kulcsot. Alapvetően a user.home könyvtárában a .keystore.
2. A parancs beütése után a következő kérdésekre kell válaszolnunk:

Enter keystore password: password
What is your first and last name?
  [Unknown]: Akos Munnich
What is the name of your organizational unit?
  [Unknown]: IK
What is the name of your organization?
  [Unknown]: DE
What is the name of your City or Locality?
  [Unknown]: Debrecen
What is the name of your State or Province?
  [Unknown]: Hungary
What is the two-letter country code for this unit?
  [Unknown]: HU
Is <CN=Akos Munnich, OU=IK, O=DE, L=Debrecen, ST=Hungary, C=HU> correct?
[no]: yes

Ezzel elkészült a kulcs. Most alá kell írnunk a JAR fájlt: jarsigner jar-neve kulcs alias neve, megadjuk a jelszót és készen vagyunk.

A keytool további opciókkal bővíthetők, a keytool –help-ben megtaláljuk ezeket az opciókat.

Fontos megjegyezni, hogyha több JAR fájlunk van, ami biztonsági kockázattal járó műveletet hajthat végre, akkor mindegyik JAR fájlra el kell végeznünk az aláírást.

3.1.4. Applet + Php

Az adatok biztonságos kezeléséhez, és a program paraméterezhetőségének megtartása érdekében nem közvetlen jdbc-s kommunikációt építünk ki az Applet és az adatbázis között, hanem egy Php fájlon keresztül történik az adatok mentése az adatbázisba. Így az Applet csak adatokat küld, adatbázis elérési funkciója nincs.

A programon belül a jatek0003.Jatek0003 osztály sendDataToPhp() metódusa foglalkozik a megvalósítással. Az Applet a JNLP-ben paraméterként megadott Php fájlnak küldi az adatokat.

3.1.5. Miért Applet?

Böngészőkbe könnyen integrálható, interaktív programokat tudunk vele létrehozni. Kényelmesen kezelhető, a Java előnyeit ki tudjuk használni vele. Széles körben elterjedt, például a CIB Bank Zrt. internet bankjának felülete is Applet. Hátránya az Appleteknek, hogy sok biztonsági akadállyal kell megbirkóznunk ahhoz, hogy a Java minden előnyét valóban ki tudjuk használni.

3.2. A program működésének bemutatása

Ebben a fejezetben a programozási háttér és a technikai megoldások általános áttekintése után, rátérünk a program konkrét működésére. Milyen csomagokkal, interfészekkel, osztályokkal valósult meg a program? Mik a program határai? Valamint bemutatjuk a programban implementált 3 játékot és a második játékhoz készített demót is.

3.2.1. A program szerkezetének bemutatása

A program szerkezetét a csomagok áttekintésével kezdjük. Először egyenként ismertetjük, hogy melyik csomagban milyen osztályok milyen feladatokat látnak el, majd kitérünk az összefüggésekre is.

A keretben csomagban két osztály található: JatekApplet, és a JatekGUI. A JatekApplet a JApplet osztály ősosztálya. A JatekGUI osztály pedig felépíti a GUI-t, ősosztálya a JPanel és implementálja az ActionListener interfészt. Az egész GUI a BorderLayout Layout Managert használja. Erre tesszük fel a menüt, gombokat, és a CENTER részre kerülnek a játékok által használt panelek.
A következő akciókat kezeli az Applet:

1. Játék betöltése fájlból
2. Játék betöltése URL-ből
3. Játék0001 példa
4. Játék0002 példa

5. Játék0002Demo
6. Játék0003 példa
7. Szakdolgozat megnyitása
8. Névjegy
9. Játék indítása
10. Inputok
11. Eredmények
12. Eredmény mentése: Excel
13. Játék bezárása

A fajlkezeles csomagban mindössze egy osztályunk van: Fajlkezelo. Feladata a properties állomány betöltése, amit a JNLP fájlban paraméterként adhatunk meg. Illetve a játékok által használt XML állományokat elemzi, hogy érvényesek-e. Főbb metódusa: acceptFile(String x).

A jatek csomag. A jatek csomagban két absztrakt osztályunk van és egy absztrakt interfészünk: GameData, JatekPanel, Jatek. Ezek írják le egy játék általános alakját. Tehát, ha egy újabb játékot szeretnénk beilleszteni ebbe a programba úgy, hogy a Jatek interfészt használja fel, akkor a következő metódusokat kell implementálnunk:

- loadData(Document d)
- playGame()
- createSaveDataFormat(File save)
- closeGame()

A GameData absztrakt osztály. Az egyes játékok - ennek az osztálynak az aktuális alosztályai- tárolják itt azokat az adatokat, amikkel dolgozunk.

A JatekPanel abszrakt osztály, a játékhoz tartozó grafikus elemeket, megjelenítéseket tartalmazza. Illetve az általa használt setPanel(String s) metódus felülírásával lehet váltani a panelok között. A showMessage(String messageType) metódussal pedig a felhasználónak tudunk üzeneteket küldeni.

A customize csomag az Appletben megtalálható Swinges elemek stílusának beállítására használható. Egyetlen osztálya a Custom. Hátterekre, szövegszínre, keretek beállítására találunk opciókat. Getter-setter metódusokat tartalmaz.

A beállításokra szolgáló properties állomány bemutatása:

szakdolgozat.welcomeLabel=Üdvözlő szöveg helye.

szakdolgozat.jatek0001Example=Játék0001 példához megadott URL.

szakdolgozat.jatek0002Example= Játék0002 példához megadott URL.

szakdolgozat.jatek0003Example= Játék0003 példához megadott URL.

szakdolgozat.szakdolgozatText= Szakdolgozat megnyitásához megadott URL.

3.2.2. Jatek0001 bemutatása

Ez a játék a 2.1. fejezetben leírt „Játék a természettel" című játék megvalósítása. Kiszámítjuk a következő stratégiák melletti legjobb választást:

- Minimax kritérium
- Maximin kritérium
- Minimax megbánás kritérium
- Maximax kritérium
- Nem elégséges ok kritérium

Illetve megadjuk, milyen α érték mellett milyen választás valószínűsíthető.

A játék a következő inputtal dolgozik:

Példa XML:


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE game SYSTEM "[http://www.___elérési
út______/jatek0001.dtd]">

<game>
        <information code="jatek0001">
                <name>Arany</name>
                <description><![CDATA[A játék leírása<br>ide
kerül]]></description>
                <author>Münnich Ákos</author>
                <date>2010.01.01.</date>
        </information>
        <input>
                <collabels>béke,háború,zűrzavar/terrorizmus</collabels>
                <rowlabels>nemesfém felhalmozás,ipari befektetés,üzemanyag
felhalmozás,informatikai beruházások</rowlabels>
                <data>{{1,5,2},{5,-2,-1},{3,8,-4},{6,3,2}}</data>   </input>
</game>
```

Először megadjuk a kódolást, illetve hogy hol található az érvényesítéshez szükséges DTD fájl. Az XML gyökér eleme a <game>. Az <information> elemen belüli elemek mindegyik játékban ugyanazt jelentik. Meg kell értelemszerűen adni a játék nevét, leírását, szerzőjét és a dátumot. Az <information> elem „code" attribútuma biztosítja a DTD-vel együtt, hogy a megfelelő XML a megfelelő leírással rendelkezzen.

Az <input> elemben megadjuk az oszlopcímkéket és a sorcímkéket. Aztán pedig a sor-oszlop értékek megadása következik szokásos formában.

A játékhoz tartozó DTD fájl:

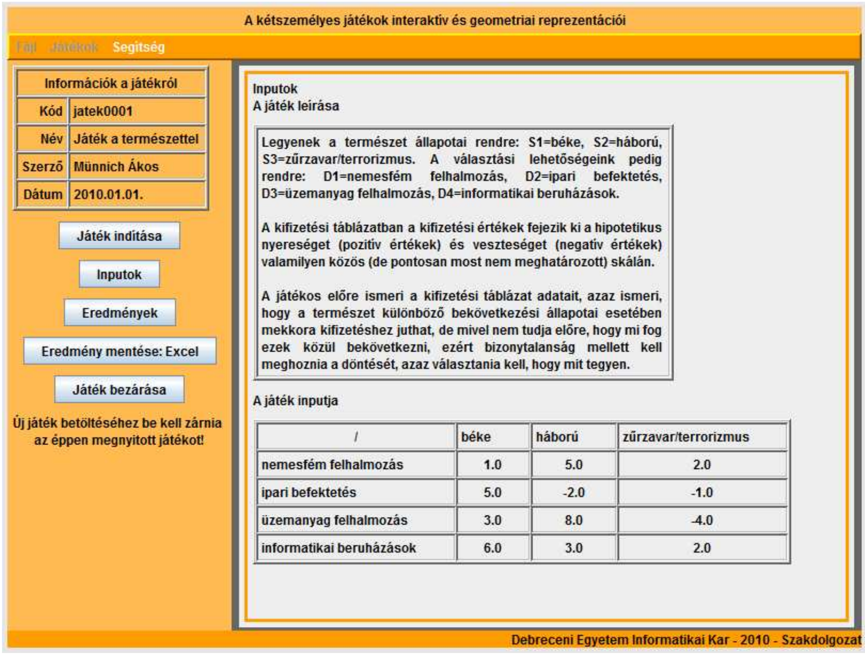
```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT game (information,input)>

<!ELEMENT information (name,description,author+,date)>
        <!ATTLIST information code CDATA #FIXED "jatek0001">

        <!ELEMENT name (#PCDATA)>
        <!ELEMENT description (#PCDATA)>
        <!ELEMENT author (#PCDATA)>
        <!ELEMENT date (#PCDATA)>

<!ELEMENT input (collabels,rowlabels,data)>
        <!ELEMENT collabels (#PCDATA)>
        <!ELEMENT rowlabels (#PCDATA)>
<!ELEMENT data (#PCDATA)>
```

Megfigyelhető, hogy a játék kódját fixen „jatek0001"-nek kell adnunk, mert így kiküszöbölhető az a helyzet, hogy a program az egyes játékok betöltésénél rossz játékot nyisson meg. Több szerzőt is megadhatunk egy-egy játékhoz.

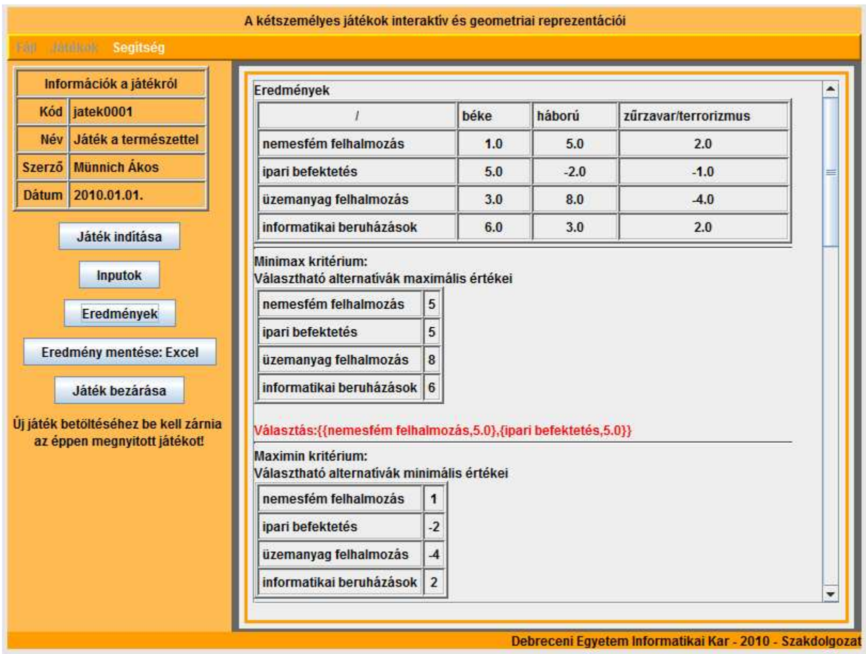
A Jatek0001 eredményeit Excel fájlba tudjuk menteni. Az eredménybe a játék információi, az input mátrix és az α értékek mellett javasolt legjobb választást leíró táblázat kerül.

Pillanatkép:

Input:



Output:



3.2.3. Jatek0002 bemutatása

Ez a játék a 2.2. fejezetben leírt „Játék értelmes ellenféllel” című játéknak a geometriai megvalósítása, a Nash-féle egyensúlypontot keressük meg és ábrázoljuk.
A játék a következő inputtal dolgozik:
Példa XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE game SYSTEM "[http://www.___elérési
út______/jatek0001.dtd]">
<game>
	<information code="jatek0002">
		<name>Családi vita</name>
		<description><![CDATA[A játék leírása<br>ide
kerül]]></description>
		<author>Münnich Ákos</author>
		<date>2010.01.01.</date>
	</information>
	<input>
		<col1>Oszlop1</col1>
		<col2>Oszlop2</col2>
		<row1>Sor1</row1>
		<row2>Sor2</row2>
		<Adata>{{1,-1},{-1,2}}</Adata>
		<Bdata>{{2,-1},{-1,1}}</Bdata>
	</input>
</game>
```

Az <input> elemben egyesével megadjuk az oszlopcímkéket és a sorcímkéket, ezzel tudjuk leginkább biztosítani, hogy helyes XML legyen az input. Mivel 2x2-es játékról van szó ez nem túl nagy pluszbefektetést jelent. A mátrixokhoz tartozó értékeket Jatek0001-hez hasonlóan adjuk meg.

A játékhoz tartozó DTD fájl:

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT game (information,input)>

<!ELEMENT information (name,description,author+,date)>
	<!ATTLIST information code CDATA #FIXED "jatek0002">

	<!ELEMENT name (#PCDATA)>
	<!ELEMENT description (#PCDATA)>
	<!ELEMENT author (#PCDATA)>
```

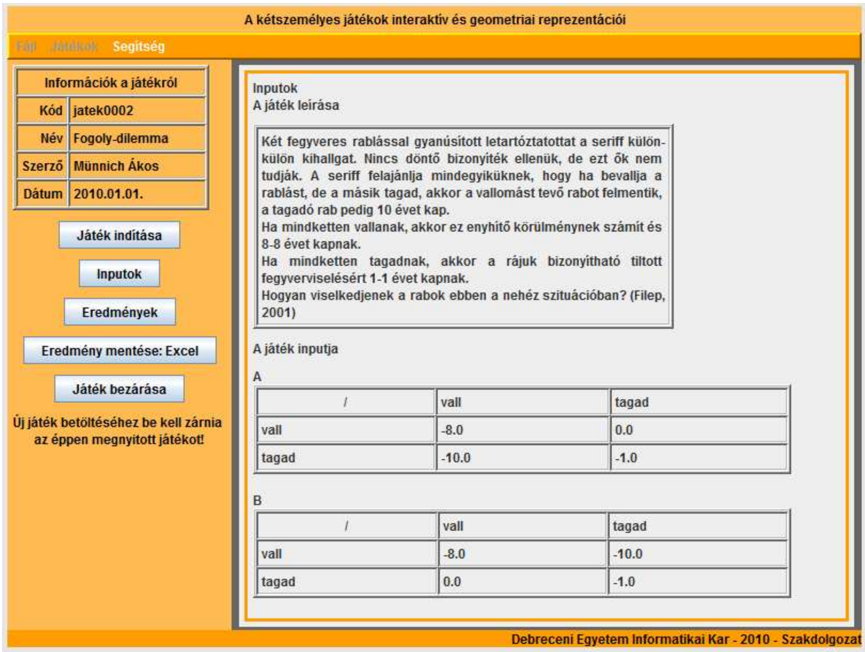
<!ELEMENT date (#PCDATA)>

<!ELEMENT input (col1,col2,row1,row2,Adata,Bdata)>
<!ELEMENT col1 (#PCDATA)>
<!ELEMENT col2 (#PCDATA)>
<!ELEMENT row1 (#PCDATA)>
<!ELEMENT row2 (#PCDATA)>
<!ELEMENT Adata (#PCDATA)>
<!ELEMENT Bdata (#PCDATA)>

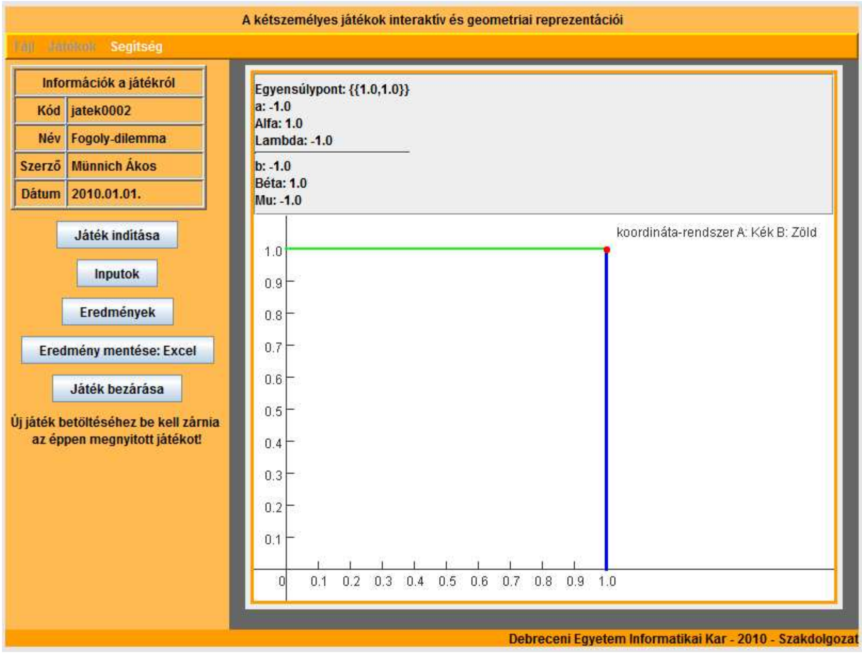
A játék kódját hasonlóan Jatek0001 esetében fixen „jatek0002” kell megadnunk. A Jatek0002 eredményeit szintén Excel fájlba tudjuk menteni. Az eredménybe a játék információi, az input mátrix és az egyensúlyi pontok koordinátái kerülnek.

Pillanatkép:

Input:



Output:

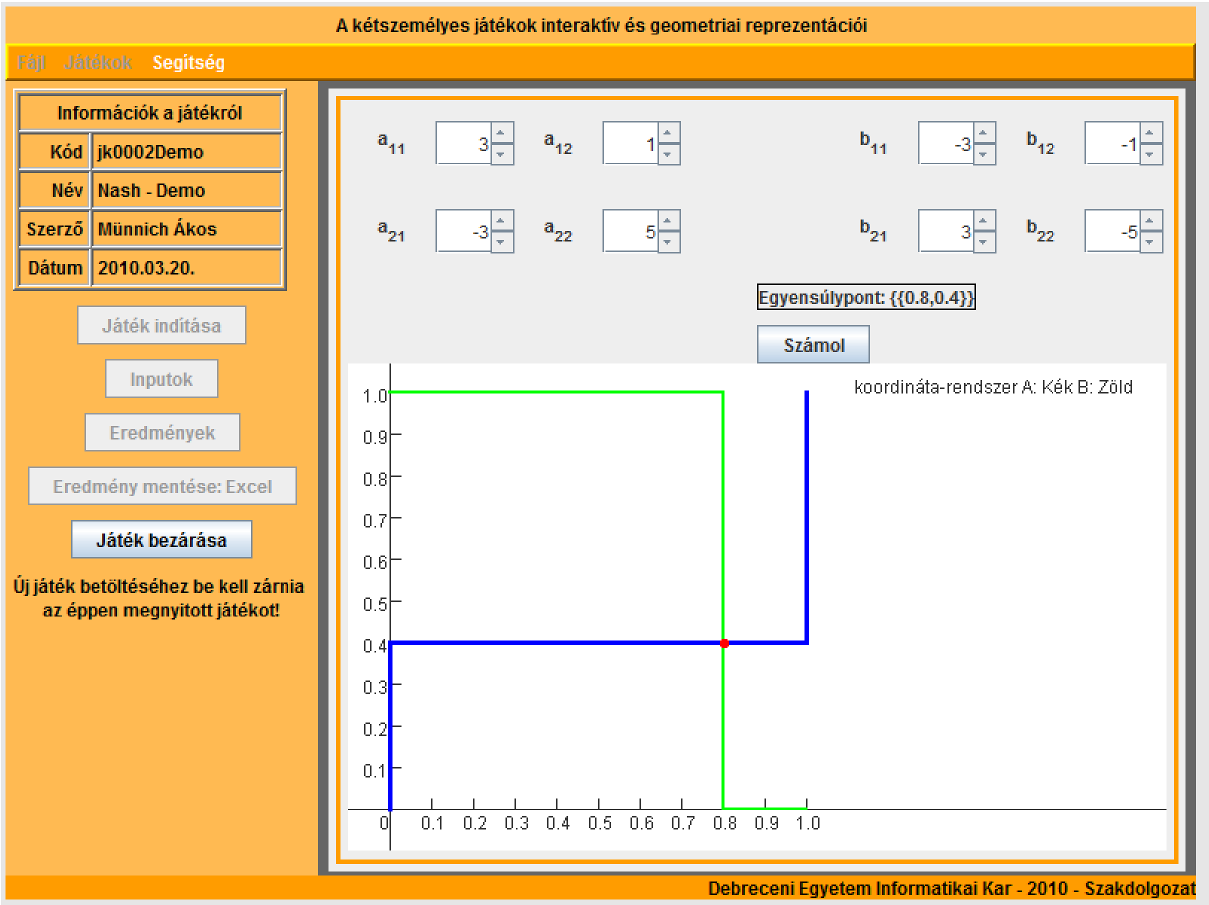


3.2.4. Jatek0002Demo bemutatása

A demó funkciójában annyiban tér el az előbb ismertetett Jatek0002-től, hogy nincs szükségünk input XML állományra, nincs szükség semminek a betöltésére. Megadunk két mátrixot rákattintunk a „Számol” gombra, és ábrázolva megkapjuk az egyensúlypontokat. Jatek0002 is, de főképp a demó verzió megvalósítása, színeiben és paramétereiben a Wolfram Research által demonstrációként létrehozott programmal egyezik meg. (http://demonstrations.wolfram.com/SetOfNashEquilibriaIn2x2MixedExtendedGames/)
A mátrix értékek csak egészek lehetnek és a [-10;10] intervallumba esnek, az „A” játékost kék színnel a „B” játékost pedig zöld színnel ábrázoljuk. Az egyensúlyi pontok pirosak. Különbözőség, hogy a demóban és Jatek0002-ben is nem törtszámként, hanem lebegőpontos számként jelenítjük meg az eredményeket.

Ennek a játéknak nem menthetjük az eredményeit Excel fájlba.

Pillanatkép:



3.2.5. Jatek0003 bemutatása

A 2.1.7. fejezetben leírt „Az optimizmus-pesszimizmus index kísérletes becslése” kísérlet alkalmazása. A eredményeket egy MySql adatbázisban tároljuk. Az adatbázisnak egy táblája van, amibe egyszerűen eltároljuk az adatokat. Ezek a következőek (2.1.7. fejezet alapján):

- S1 – másik jelöléssel, amit az input XML használ E1
- S2 – E2
- D1 – A1
- D2 – A2
- adott x érték
- adott x érték melletti választás
- életkor
- nem
- időpont

A játék a következő inputtal dolgozik:

Példa XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE game SYSTEM "[http://www.___elérési
út______/jatek0001.dtd]">

<game>
	<information code="jatek0003">
		<name>Teszt</name>
		<description><![CDATA[A játék leírása<br>ide
kerül]]></description>
		<author>Münnich Ákos</author>
		<date>2010.04.01.</date>
	</information>
	<input order="up">
		<E1>Béke</E1>
		<E2>Háború</E2>
		<A1>Civil ipar</A1>
		<A2>Olaj</A2>
		<step>0.1</step>
	</input>
</game>
```

Az <input> itt valamivel eltér az előzőektől. Az „order” attribútumban meg kell adnunk, hogy a kísérlet során a kérdéseket milyen sorrendben kapja a tesztet kitöltő személy. Ez lehet: up, down, random. A címkéket szintén egyesével adjuk meg. A <step> elem pedig a kísérlet finomságára vonatkozik. A tesztkérdések x elemei [0;1] intervallumban vesznek fel értékeket.

A játékhoz tartozó DTD fájl:

```
<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT game (information,input)>

<!ELEMENT information (name,description,author+,date)>
    <!ATTLIST information code CDATA #FIXED "jatek0003">

    <!ELEMENT name (#PCDATA)>
    <!ELEMENT description (#PCDATA)>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT date (#PCDATA)>


<!ELEMENT input (E1,E2,A1,A2,step)>
    <!ATTLIST input order (up|down|random) "random">

    <!ELEMENT E1 (#PCDATA)>
    <!ELEMENT E2 (#PCDATA)>
    <!ELEMENT A1 (#PCDATA)>
    <!ELEMENT A2 (#PCDATA)>
    <!ELEMENT step (#PCDATA)>
```

A játék elindítása után a tesztalany minden egyes kérdésre 3 lehetőség közül választhat:

- D1
- D2
- Indifferens

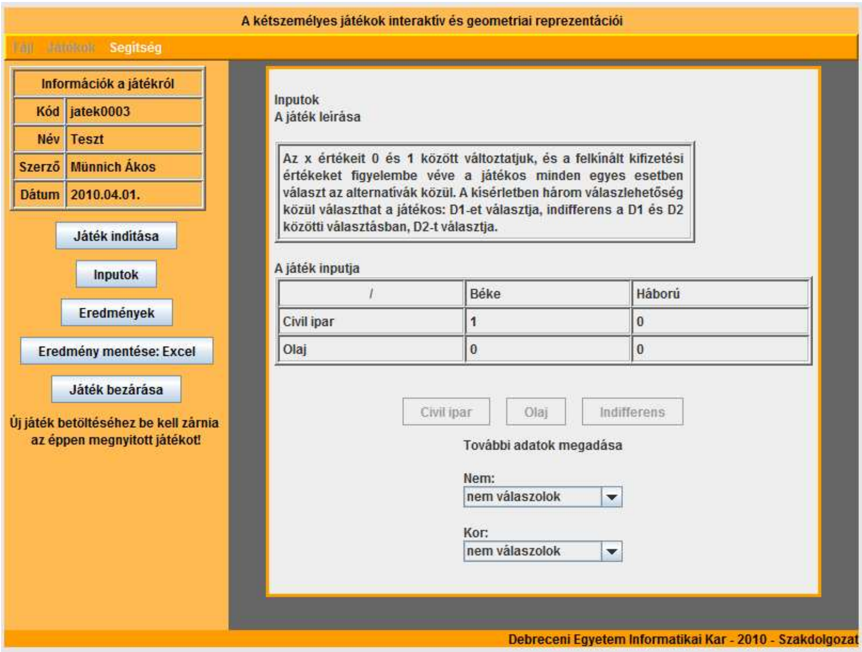
Továbbá opcionálisan megadhatja az életkorát és nemét. A teszt végén egy üzenetet kap, hogy a kísérlet véget ért. Megtekintheti saját eredményét és Excel formátumban el is mentheti azokat, a játékhoz tartozó információkkal együtt.

Látható, hogy a program innen több irányba fejleszthető tovább. Ennek a dolgozatnak azonban nem célja ezeknek az utaknak a keresése, mindösszesen a benne rejlő potenciált akarja megmutatni. Ahhoz, hogy ezeket az irányokat meghatározzuk, mély pszichológiai elemzésekre van szükség. A programban megtalálható kezdetleges űrlap, ahol a tesztalanytól bekérjük a nemét és életkorát, szintén azért nem egy különálló oldalon lett megvalósítva, mert jelenleg nem tudjuk pontosan, hogy milyen adatokra is lesz szükségünk. Csak jelezzük, hogy egy űrlapra szükség lehet. Továbbá nem szűrjük ki már az adatok elküldése előtt, hogy a kitöltött teszt eredményei később felhasználhatóak-e vagy sem, mivel nem tudjuk jelenleg ezt meghatározni.

Most tehát csak annyi történik, hogy mindent, amit csak lehet, adatbázisban tárolunk és ezeket az eredményeket elemzés céljára felhasználhatóvá teszünk.

Pillanatkép:

Kísérlet közben:



Output:

3.2.6. A program felhasználói felülete

A program szerkezetének és a megvalósított játékoknak a bemutatása után foglalkozzunk a felhasználói felülettel.

A címsorba a szakdolgozat címe található: A kétszemélyes játékok interaktív és geometriai reprezentációi.

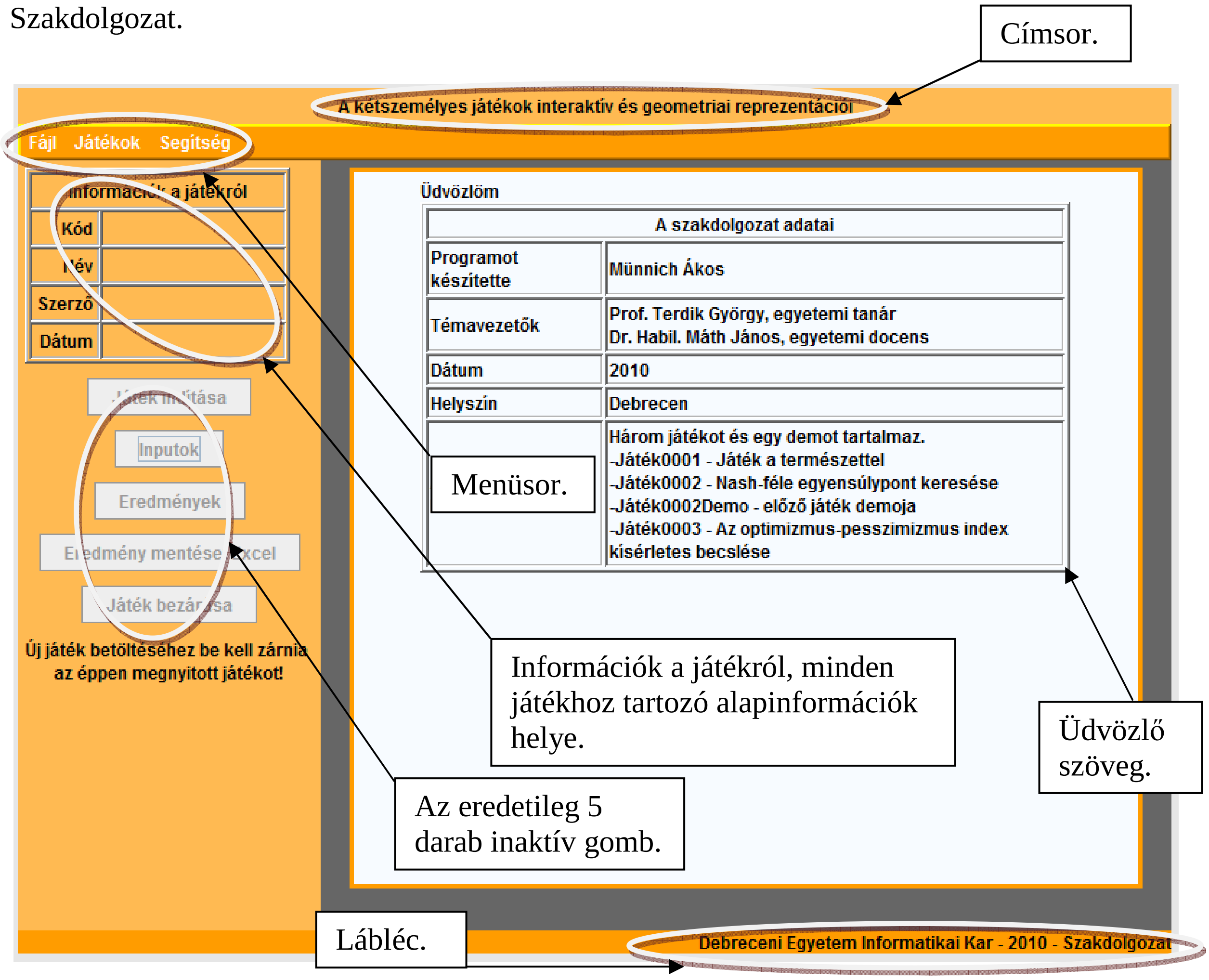
A menüsorban a következő szerkezetben érjük el a funkciókat:

1. Fájl
   a. Játék betöltése fájlból
   b. Játék betöltése URL megadással
2. Játékok
   a. Játék0001 példa
   b. Játék0002 példa
   c. Játék0002Demo
   d. Játék0003 példa
3. Segítség
   a. Szakdolgozat megnyitása
   b. Névjegy

A játék betöltése után a bal oldalon található táblázatban, a játékhoz tartozó alapinformációkat olvashatjuk, a játék betöltése után. Alatta 5 gomb van. Ezek aktivizálódása az egyes játékok megvalósításától függ. Jelen esetben a Jatek0002Demo (Nash-egyensúlypont geometriai reprezentációja) játék kivételével mindegyik aktivizálódik.

A középső panelra töltődnek be a játékok, alapállapotban egy üdvözlőszöveg látható, amit a properties fájlból töltünk be.

A láblécre pedig a következő felirat került: Debreceni Egyetem Inromatikai Kar – 2010 – Szakdolgozat.



3.2.7. Egy játék betöltésétől a bezárásáig

Amikor egy játékot be szeretnénk tölteni, két lehetőségünk van:

1. Fájl/Játék betöltése fájlból: ekkor egy FileDialog ablak ugrik fel, ahol ki tudjuk választani az adott XML állományt. A kiválasztás után ismét két eset van:
    a. A kiválaszott XML-t a program validálja és átkerül a vezérlés a játékhoz.
    b. Hibás formátumú fájlt választottunk ki. Ekkor hibaüzenetet kapunk és nem történik meg a betöltés.
2. Fájl/Játék betöltése URL megadással: egy szövegmezőbe megadjuk az XML-hez tartozó URL-t, rákattintunk a „Játék betöltés URL-ből” gombra, és 1. pont a. vagy b. esetével folytatódik a program.
3. Amennyiben a FileDialog ablakban a „Mégse” opciót választjuk, nem töltődik be játék, és egy üzenetet kapunk, miszerint nem választottunk ki fájlt.

Miután sikeresen betöltöttünk egy játékot, a menüsorban a „Fájl” és a „Játékok” menüpont inaktívvá válik, ezzel biztosítva, hogy egyszerre csak egy játékkal tudjunk foglalkozni. Kitöltődik a baloldali táblázat és az eddig inaktív gombok a játéktól függően aktivizálódnak.

Bármely gombra kattinthatunk, de amíg a Játékot nem játsszuk le, addig az „Eredmények” pont alatt nem fogunk eredményeket látni. Alaphelyzetben az „Input” panelre kerülünk. Ha egyből el szeretnénk menteni a nem létező eredményeket, akkor hibaüzenetet fogunk kapni, hogy az eredmény addig nem menthető, míg a játék be nem fejeződött.
Miután a „Játék indítása” gombra kattintunk a Jatek0001 és a Jatek0002 esetében az eredményeket már meg is tekinthetjük, mivel a számítások elvégzése után nincs több teendőnk az eredmények elemzésén kívül. Jatek0003 esetében amíg a kísérlet nem fejeződik be, addig az eredmények pont alatt nem találunk outputot és Excel formátumba sem tudunk menteni.
Ahhoz, hogy új játékot töltsünk be, szabályosan be kell zárnunk az előző, éppen megnyitott játékot. Ezt úgy tudjuk megtenni, hogy a „Játék bezárása” gombra kattintunk. Ekkor újra a kezdő oldalra kerülünk, és aktívvá válik a „Fájl” illetve „Játékok” menüpont. A program alapállapotba kerül.

3.2.8. Üzenetek a felhasználó felé

Ebben a rövid részben áttekintjük, hogy a program milyen üzenetekkel kommunikál a felhasználóval.

- Hibás formátumú fájl! – Amennyiben a kiválasztott fájl nem megy át a validáláson.
- Nem választott ki fájlt! – Ha a fájl betöltésekor úgy döntünk, hogy a „Mégse” opciót választjuk, erről értesítést kapunk.
- Sikeres betöltés. Kattintson az OK gombra! – Amennyiben a betöltés sikeres.
- Az eredményeket addig nem lehet elmenteni, amíg a játék véget nem ért! – Akkor kapjuk, amikor úgy próbálunk menteni, hogy még nincsenek meg az eredmények. Miután tudomásul vesszük a problémát, kapunk még egy olyan üzenetet is, ami jelzi, hogy nem történt mentés.
- A mentés nem hajtható végre. – Ha hiba történik a mentés közben, vagy ha úgy próbálunk menteni, hogy arra még nincs lehetőségünk.
- Nem sikerült az adatokat elmenteni az adatbázisba! – A teszt kitöltése után az adatok egy adatbázisba kerülnek, ha valamiféle hiba lép fel, akkor ezt az üzenetet kapjuk.
- Az adatokat sikeresen mentettük! – A teszt kitöltése után az adatok sikeresen az adatbázisba kerültek.

4. Összefoglalás

Ahogy a bevezetőben megfogalmaztuk, a játékelmélet felhasználása mára nagyon széles körben elterjedt, aminek áttekintése meghaladja ennek a dolgozatnak a kereteit. Emiatt is a dolgozat legfőbb célja az volt, hogy a játékelméletben rejlő, a gyakorlatban is hasznos és egyben érdekes lehetőségeket bemutassa.

Néhány alapkoncepcióból indultunk ki, amit a „Kétszemélyes játékok játékelméleti alapjai” fejezetben részletesen ismertettünk. Bemutattuk, hogy milyen módon, milyen stratégiák mentén hozhatunk leginkább racionális döntéseket. A döntési helyzetektől függően viszont be kell látnunk, hogy az egyes individuumok másképp viselkednek. Nem minden esetben hozunk racionális döntéseket, sokkal inkább az érzelmeinkre hallgatunk.

Ennek megfelelően foglakoztunk az optimizmus-pesszimizmus index-szel (Hurwicz, 1951), illetve megalkottunk egy kísérletet. A teszt segítségével egy egyszerű becslést tudunk adni arra, hogy egy adott egyén milyen optimizmus-pesszimizmus értékkel rendelkezik. Ezt a tudást felhasználva pedig további következtetések levonására nyílik lehetőségünk. Az adatbázisban tárolt életkor és nem alapján például megbecsülhetnénk, hogy egyes korcsoportok a nemek eloszlása mellett, milyen optimizmus-pesszimizmus értékkel rendelkeznek például, ha kedvenc focicsapatuk bajnoki esélyeiről, vagy ha az ország sorsáról kérdezzük őket.

A kísérletben rejlő további lehetőségek kiaknázására részletes, főképp pszichológiai elemzésekre van szükség. Ezután lehet nekilátni a teszt finomításának és az adatok felhalmozásának.

Lényeges szerepet kapott a dolgozatban a 2x2-es játékok elemzése és grafikus megjelenítése. Ebben a részben leginkább Filep (2001) tankönyvét vettük alapul. A fő hangsúlyt a definíciók és a tételek ismertetése után az eredmények megjelenítésére fektettük. Nem mentünk bele a matematikai háttér részletes tárgyalásába. Ennek a fejezetnek a jelentősége leginkább az oktatásban érvényesülhet, ahol a megszerzett elméleti ismeretek után és mellett, könnyen demonstrálhatóvá válnak az egyes problémák és azok lehetséges megoldásai, valamint bárki készíthet magának saját feladatokat, amit be tud tölteni a programba és felhasználhatja annak eredményeit.

Összességében megállapíthatjuk, hogy egy könnyen kezelhető, a problémáinkra gyors és egyértelmű választ adó programot sikerült létrehozni, melynek eredményeit egyéb informatikától távolabb álló ágazatokban is fel lehet használni.

5. Irodalomjegyzék

Canty, M. J. (2003). *Resolving Conflictswith Mathematica®*. New York: Academic Press

Filep, L. (2001). *Játékelmélet*. Budapest: Filum Kiadó

French, S. (1986). *Decision Theory: an introduction to the mathematics of rationality*. Chichester, West Sussex: Ellis Horwood Limited

Hurwicz, L. (1951). Optimality criteria for decision making under ignorance. *Cowles Commission Discussion Paper.* No. 370 (mimeographed)

Kelly, A. (2003). *Decision Making using Game Theory: An Introduction for Managers*. Cambridge: Cambridge University Press

Laplace, P. S. (1825). *Essai Philosophique sur les Probabilites.* 5th edition, Paris. Translation published by Dover, New York

Luce, R. D. and Raiffa, H. (1957). *Games and Decisions: Introduction and Critical Survey*. New York: John Wiles and Sons, Inc.

Mészáros, J. (2003). *Játékelmélet*. Budapest: Gondolat Kiadó

Nash, J. F. (1951). Non-cooperative games, *Ann. of Math.* 54, 286-295.

Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behaviour*. Princeton, NJ: Princeton University Press

Osborne, M.J. and Rubinstein, A. (1994). *A Course ín Game Theory.* Cambridge: The MIT Press

Savage, L. J. (1951). The theory of statistical decision. *J. of Amer. Statist. Assoc.* 46, 55-67.

Tóth, I. J. (1997). *Játékelmélet és Társadalom*. Szeged: JATEPress

Szidarovszky F. (1977). *Játékelmélet*. Budapest: Tankönyvkiadó

Wald, Á. (1950). *Statistical Decision Functions.* New York: John Wiley

Zagare, F. C. (1984). *Game Theory: Concepts and Applications*. Becerly Hills and London: Sage Publications, Inc.

http://java.sun.com/applets/

http://java.sun.com/docs/books/tutorial/deployment/applet/index.html

http://java.sun.com/docs/books/tutorial/deployment/applet/deployingApplet.html

http://demonstrations.wolfram.com/SetOfNashEquilibriaIn2x2MixedExtendedGames/

http://www.ezit.hu/?cat=158