

Tárgy:

BPI1235L *Mesterséges intelligencia, online verzió*

Kreditpont : 4

Féléves kontakt óraszám (elm.+gyak.): 9+9

Előfeltétel (tantárgyi kód) BPI1204L (programozás 2)

Oktató: dr. Vályi Sándor

- elektronikus *levelező*cím: valyi.sandor@nye.hu, 3 munkanapi reakcióidő, nem azonnali ügyfélszolgálat!
- A kurzushoz tartozik: moodle.nye.hu (Kezdőoldal / ► Matematika és Informatika Intézet / ► Informatika / ► PTI III. év levelező / ► Mesterséges intelligencia). Itt lesznek az előadások anyagai.
- A kurzushoz tartozik: classroom.google.com- Felvételi kulcs: dmgss6l itt fogunk gyakorló feladatokat megoldani interaktívan.
- A kurzushoz tartozó *MSTeams* csoport: MestintLEV

Félévi követelmény: vizsga (évközi alapján lehetséges jegymegajánlás is)

Évközi követelmény, vizsgára bocsájtás feltétele:

online beadandó-védés és online dolgozat+szóbeli felelet

Online módszerek és követelmények:

Az előadások menete: a prezentációk a **moodle**-kurzusnál vannak elhelyezve. **Az MS Teams**-en lesz az előadás, ott képernyő-megosztással beszél az oktató. A felvételt a Teams-en pár héten keresztül újra meg lehet tekinteni, letölteni -- .a további megosztást viszont az oktató nem engedélyezi. Az előadások az órarendnek megfelelő időpontokban zajlanak.

A gyakorlatokon más a helyzet: ott a **classroom**-on kapnak gyakorló feladatokat, amiket ott is kell benyújtaniok, és én szelektíven át is nézi az oktató a megoldásokat, de csak tanácsadási, nem ellenőrzési, pontozási céllal. Ezzel párhuzamban az **MSTeams**-en fog menni egy találkozó. A beadandókról való konzultáció szintén ezekben az időpontokban fog menni.

Évközi követelmények, vizsgára bocsájtás feltételei:

1. A PTI-hallgatók egy MI feladatot számítógépes **program és dokumentáció készítésével** megoldanak és dokumentálva benyújtanak a moodle-re feltöltve vagy a github-on. Ez 40 pont. Határidő: az utolsó évközi gyakorlat. MSTeams-értekezleten kell megvédeni, ahol azt vizsgálom, hogy érti-e a programkódot a benyújtó.

2. **classroom-on** megírt **online dolgozat**, az utolsó évközi gyakorlaton. Ez 60 pont. Ennek szóbeli védésével, ún. feleléssel lehet ténylegesen megszerezni a pontokat.

Mindkét évközi követelményből 58%-ot kell szerezni a vizsgázás lehetőségéhez. Amennyiben valamelyik évközi követelmény nem sikerül, úgy ez(eke)t 1x lehet javítani, a vizsgaidőszak hozzátvetőlegesen közepén (június elején).

Az évközi teljesítmény alapján jegymegajánlás kapható, 58p: elégséges, 70p: közepes, 80p: jó, 90p: jeles.

Vizsga: Ha egy vizsgát felvesz a hallgató, úgy, hogy nincs meg az évközi követelmény, úgy elégtelent fogok beírni. Az el nem fogadott jegymegajánlás pontveszejtő hatású, a szóbeli vizsgán mindenféle eredmény elérhető az évközi pontoktól függetlenül, bár persze nagyon is függően az ott megszerzett tudástól.

A beadandóról:

A beadandó program **problémamegoldója** az **állapottér gráfrepresentációján** kell alapuljon és valamely, a kurzuson **tanult keresőalgoritmust** felhasználva (mélységi, szélességi kereső etc. avagy minimax algoritmus, alfa-béta vágás) kell működni. **[20p]**

1-személyes játék esetén kezdődjön **pályaszerkesztővel**, ha van értelme, 2-személyes játék esetén is kezdődjön pályaszerkesztővel (**állászerkesztővel**). 1 személyes játék esetén a megszerkesztett feladatra fusson a kereső eljárás, és lépésenként mutassa be a megtalált megoldást. 2 személyes játék esetén az egyik ellenfél lépéseit humán vezesse, a másik fél lépéseit a minimax lépésajánló, legalább 8 féllépés mélységig.

Mindkét esetben a beadandó program **grafikus felhasználói felülettel** rendelkezzen. **[10p]** Legalább **annyi távoli adatbázis-használatot** tartalmazzon, hogy elmentse az adott felhasználó nevét, a megszerkesztett generált pályát, a játszmát és az eredményt, vagy relációs adatbázisban, vagy XML-adatként. Mondani sem kellene, vissza is kell tudnia tölteni a mentésből. **[5p]**

A megoldás **elkészítése előtt** a tanárral való **egyeztetés szükséges** a megírandó programmal szemben támasztott **követelményekről**. Ennek módja a gyakorlatokon való személyes megjelenés. A program megoldó algoritmus az állapotter-reprezentáción kell alapuljon, s valamelyik, az előadáson vett kereső-algoritmust kell megvalósítania.

- A bemutatók feladatokban való mély ismereteikről **személyes védés** során számolnak be. Az elkészített programban meg kell jelölni azon kódrészeket, amelyeket nem a védő készített. Olyan kérdések is várhatók, hogy mit kellene másképp csinálni, ha változtatni akarnánk a program működésén. Amennyiben a védés nem sikeres, a vizsgázást nem engedélyezem.

- A **dokumentáció** tartalmazza az állapotter egy 10-12 állapotú részének ábráját, ahol a lehetséges operátorok is elmagyarázhatók, valamint azt, hogy milyen kereső lett leprogramozva. A program az állapotterében való keresés egyes lépéseit (az érintett csúcshalmazokat és az alkalmazott műveletet) is tudja kiírni. **[10p]**

Kivétel a fent leírt beadandó-követelmények alól:

- aki tanuló ágenst készít open source könyvtárral adott inputból adott intervallum-értékeket előállító intervallum-értékű számítások előállítására, pl. 'Deep Learning'

neuronhálókat alkalmazva, ő jeles megajánlott jegyet kap, s sejtetően mehet vele országos TDK-ra is.

- aki működő ágenst készít reális (amely problémákat szokták mellékelni PC-s sokobanhoz) Sokoban feladatok megoldására, jeles megajánlott jegyet kap.
- aki olyan webhelyet készít, Java web service, Spring MVC (ez a preferált!) technológiára, amelyre a regisztrált userek felvihetnek Sokoban álláskiértékelő heurisztikát visszaadó int heur(SokobanState) metódus szövegét, Java-ban megírva, és a webhely eme heurisztikát injektálva az A* keresőbe keres megoldást a szintén a userek által felvitt pályákra, és időszakonként rangsort hirdet a userek heurisztikái között, a teljesítményük alapján, ő jeles megajánlást kap.

Tananyag:

Alkalom	Előadás jellegű	Gyakorlati jellegű	
1	Bevezetés, intelligens ágens	Ágens tulajdonságai; Egyszerű ágens programozása.	
1	Problémák állapotér-reprezentációja	Példák állapotér-reprezentációra, keretrendszer állapotér-reprezentált problémák számára	
1	Problémamegoldás, mint útkeresés	Mélységi, szélességi és egyéb nem informált keresők.	
1	Informált keresési eljárások	Heurisztikák vizsgálata. Best-first, A* végrehajtása.	
2	2-személyes játékok	Minimax algoritmus, alfa-béta vágás.	
2	Példa a beadandóhoz.	Keretrendszer a 2-személyes játékokhoz.	
2	Lokális keresők, szimulált hűtés. Genetikus algoritmusok. Backtrack algoritmusok.		
2	Logikai ismeretreprezentáció, tételbizonyító algoritmus (rezolúció), logikai programozás. Kényszerfeltételes logikai	logikai programozás, kényszerfeltételes logikai programozás: példák.	

	programozás.		
3		Classroom alapú dolgozat az elméleti és gyakorlati részekből. (anyag: Ágensek, állapottér-reprezentáció, útkereső algoritmusok, 2-személyes játékok, minimax, alfabéta-vágás).	
3	A dolgozat szóbeli védése MSTeams-en		
3		A beadandók védése MSTEamsen.	

Oktatási segédanyag:

Az előadáson bemutatott prezentációk megtalálhatók a <http://moodle.nye.hu> címen.

Kötelező¹ és ajánlott irodalom:

- Futó Iván (szerk.): Mesterséges intelligencia, Aula Kiadó, 1999.
- Fekete István, Gregorics Tibor, Nagy Sára: Bevezetés a mesterséges intelligenciába, LSI Oktatóközpont, 1990, ELTE Eötvös Kiadó, 2006.

Stuart J. Russell, Peter Norvig¹: Mesterséges intelligencia modern megközelítésben, Panem, Budapest, 2005. (1-9. fejezet a kötelező, a 2003-as második kiadás fordítása) ,
Online elérhető, URL:

<https://dtk.tankonyvtar.hu/handle/123456789/8703>