

**Tárgy:** PMB1218L *Mesterséges intelligencia*

**Oktató:** Vályi Sándor

**Meghirdetés féléve:** 2018. ősze

**Kreditpont :** 4

**Heti kontakt óraszám (elm.+gyak.):** 13+13

**Félévi követelmény:** kollokvium

**Előfeltétel (tantárgyi kód)** PMB1205 (programozás 2)

**Évközi követelmények, vizsgára bocsájtás feltételei:**

A PTI-hallgatók egy MI feladatot számítógépes program készítésével megoldanak és dokumentálva benyújtanak. Ez 40 pont. **Határidő: az utolsó évközi gyakorlat.**

**Vizsga:** Papíron megírt vizsga, elméleti és gyakorlati feladatokkal, ez utóbbiak között programozási feladat is. Ez a vizsga 60 pont.

Az elégségeshez 50 pont kell, az évközi teljesítményből (program, 40p) és a papíron írt vizsgából (60p) összegezve. Minden 10% javulás egy jegy javulást jelent, maximum ötösig.

A feladat kiválasztása a tanárral e-mailben lehetséges

A program az **állapottér gráfrepresentációján** kell alapuljon és valamely, a kurzuson **tanult keresőalgoritmust** legalább 2-féle módon felhasználva (mélységi, szélességi kereső etc. avagy minimax algoritmus, alfa-béta vágás) kell működnie.

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. Legalább **annyi távoli adatbázis-használatot** tartalmazzon, hogy elmenti az adott felhasználó nevét, a megszerkesztett generált pályát, a játszmát és az eredményt, vagy relációs adatbázisban, vagy XML-adatként.

A megoldás **elkészítése előtt** a tanárral való **egyeztetés szükséges** a megírandó programmal szemben támasztott **követelményekről**. Ennek módja a gyakorlatokon való személyes megjelenés. Az egyeztetés eredményét előzetes kis dokumentációban rögzítjük a tanár és a hallgató között. A program megoldó algoritmus az állapottér-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 állapottér egy 10-12 állapotú részének ábráját, ahol a lehetséges operátorok is elmagyarázhatók, valamint azt, hogy milyen kereső lett leprogramozva. A program az állapottérben való keresés egyes lépéseit (az érintett csúcshalmazokat és az alkalmazott műveletet) is tudja kiírni.

**Kivétel a fent leírt beadandó-követelmények alól:**

- aki tanuló ágenszt készít open source könyvtárral adott inputból adott intervallum-értékeket előállító intervallum-értékű számítások előállítására, pl. 'Deep Learning' neuronhálókat alkalmazva. Neki erre a beadandóra kapható pontjainak száma 90p.
- aki működő ágenszt készít reális Sokoban feladatok megoldására. Neki erre a beadandóra kapható pontjainak száma 90p.
- aki olyan webhelyet készít, Apache+PHP, Node.js vagy Apache Tomcat, Spring MVC (ez a preferált!) technológiára, amelyre a regisztrált users felvihetnek Sokoban álláskiértékelő heurisztikát visszaadó `int heur(SokobanState)` metódus szövegét, Java-ban megírva, és a webhely Java Reflection-szerűen eme heurisztikát injektálva az A\* keresőbe keres megoldást a szintén a users által felvitt pályákra, és időszakonként rangsort hirdet a users heurisztikái között, a teljesítményük alapján.

**Tananyag:**

Hét	Előadáson	Gyakorlaton	
1	Bevezetés, intelligens ágensek	Ágensek tulajdonságai; Egyszerű ágensek programozása	
1	Problémák állapottér-reprezentációja	Példák állapottér-reprezentációra, keretrendszer állapottér-reprezentált problémák számára	
1	Problémamegoldás, mint útkeresés	Mélységi, szélességi és egyéb nem informált keresők	
1	Informált keresési eljárások	Informált keresők	
2	2-személyes játékok	Minimax algoritmus, alfa-béta vágás. Keretrendszer a 2-személyes játékokhoz. Példa a beadandóhoz.	
3	Genetikus algoritmusok, kényszerfeltételekkel definiált problémák megoldása. Logikai ismeretreprezentáció,	Logikai ismeretreprezentáció Tarski világában és a wumpuszvilágban. Prolog programok írása, használata Java-	

	logikai programozás.	ból.	
<b>4</b>	Szakértői rendszerek, bizonytalan információ ábrázolása. Tanuló algoritmusok.	<b>A beadandók védelése.</b>	

Oktatási segédanyag:

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

Kötelező<sup>1</sup> 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 Norvig<sup>1</sup>: Mesterséges intelligencia modern megközelítésben, Panem, Budapest, 2005. (1-9. fejezet a kötelező, a 2003-as második kiadás fordítása) ,**  
Online elérhető, URL:

[http://www.tankonyvtar.hu/hu/tartalom/tamop425/0026\\_mi\\_4\\_4/index.html](http://www.tankonyvtar.hu/hu/tartalom/tamop425/0026_mi_4_4/index.html)