

Tárgy: PMB1218 *Mesterséges intelligencia*

*Oktató: dr. Vályi Sándor*

**Meghirdetés féléve : 5**

**Kreditpont : 4**

**Heti kontakt óraszám (elm.+gyak.): 2+2**

**Félévi követelmény: kollokvium**

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

Évközi követelmények, vizsgára bocsájtás feltételei:

- Évközben a hallgatók egy **MI feladatot** számítógépes **program készítésével** megoldanak és **dokumentálva** benyújtanak. A feladat kiválasztása a <http://moodle.nyf.hu> megfelelő kurzusfelületén (beadandó feladat választása) keresztül lehetséges, a 6. gyakorlat időpontjáig, A <http://moodle.nyf.hu-ra> lehet a megoldásokat is feltölteni. Állapottér-reprezentáció beadása szintén a 6. gyakorlatig, utána a lenti táblázatban vannak leírva az beadási fázisok, végső beadás a 13. héten.

- A feladatok típusa:

*1-személyes játékban (problémamegoldás) pályaszerkesztő és automatikus megoldó, amely humán megoldóknak is komoly problémát okozó méretű problémapéldányokat is megold 10 másodpercen belül.*

*2-személyes játékban pályaszerkesztő és lépésajánló, amely 100 esetből 75-ben legyőzi a véletlen lépésgenerálással megírt automatikus ellenfelet, véletlenszerűen generált „küzdőtéren”*

*Jelesért a következő feladattal is meg lehet próbálkozni: olyan webszájt létrehozása, ahová hallgatók Neptun-nevükkel regisztrálnak, s adott 2-személyes játék heurisztikus függvényét feltölthetik. A véglegesítés és a határidő letelte után a feltöltött heurisztikákat a webszájt véletlen pályákon bajnokságban használja, és a feltöltött heurisztikák között sorrendet, pontszám-eredményeket állapít meg. (JSP+Java technológia előnyben, PHP is mehet, a megoldás szakdolgozatnak is alkalmas lehet)*

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

Mindkét esetben a beadandó program **grafikus felhasználói felülettel** rendelkezzen. Legalább **annyi távoli adatbázis-használatot** tartalmazzon, hogy elmenti az adott felhasználó nevét, a 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 algoritmusa 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 állapotterében való keresés egyes lépéseit (az érintett csúcshalmazokat és az alkalmazott műveletet) is tudja kiírni.

•A **beadott programon kívül** a két dolgozat lesz a később jövő táblázatban megadott dátumokon az *előadásokon*.

Az évközi teljesítmény **értékelése**: a beadandó program követelményeknek megfelelő véde – beugrónak számít. Az előadáson tanult állapotér-reprezentáció és keresőeljárások alkalmazási tudása, beleértve a minimax algoritmust és az alfa-béta vágást – ez lesz az első dolgozatban. Az első dolgozat eredménye 70%-ot jelent az évközi szereshető pontokból. Még 30%-ot lehet elérni a *második zárthelyin*, illetve *egyéni munka vállalásával*: az MI olyan részterületének átnézésével, amelyet az előadás nem vagy csak kevésbé érintett. Mit kell csinálni? Az átnézés után 4-5 oldalas összefoglalót készíteni, az adott terület történetével és legújabb eredményeivel és irodalomjegyzékével, és az adott terület egy szabadon elérhető szoftvere bemutatásával. Területek pl.

- Beszédfelismerés, karakterfelismerés, arcfelismerés – itt a szabadon elérhető szoftverek használatának demonstrálása a cél
- Spamszűrés, hálózati log-fájlokból támadás-kiszűrés
- Szakértői rendszerek különböző tudásterületekre
- Legó-robotok érzékelőinek használata *nem csak adatmentés, hanem intelligens használat!*
- Mobiltelefonok érzékelői adatainak feldolgozása (kamera, GPS ill. egyéb helyérzékelés, mikrofon, gyorsulásmérő) *nem csak adatmentés, hanem intelligens használat!*

Egy-egy téma lefoglalása a tanárnál személyesen lehetséges, csak az első jelentkező dolgozatát fogadjuk el. A jelentkezések listája a moodle-ban lesz megjelenítve, egy listában. Aki a két dolgozattól elér 50%-ot, az elégséges(2) megajánlást szereshet, minden +10% egy jegy javítást, egészen 5-ig.

A vizsgaidőszakban a bukott évközi követelményeket a TVSZ szerint egyszer lehet javítani, ami a program védeését jelenti, a papír dolgozat a vizsgán megírandó ebben az esetben.

**Tantárgyi program:**

**Hét Előadáson**

**Gyakorlaton**

**Dátum**

<b>1</b>	Bevezetés, intelligens ágensek	Ágensek tulajdonságai; Egyszerű ágensek programozása	<b>09.08</b>
<b>2</b>	Problémák állapottér-reprezentációja	Példák állapottér-reprezentációra, keretrendszer állapottér-reprezentált problémák számára	<b>09.15</b>
<b>3</b>	Problémamegoldás, mint útkeresés	Mélységi, szélességi és egyéb nem informált keresők	<b>09.22</b>
<b>4</b>	Informált keresési eljárások	Informált keresők	<b>09.29</b>
<b>5</b>	2-személyes játékok	Minimax algoritmus, alfa-béta vágás. Keretrendszer a 2-személyes játékokhoz	<b>10.6</b>
<b>6</b>	Genetikus algoritmusok, kényszerfeltételekkel definiált problémák megoldása	Beadandó 1. produktum: a választott témák problémáinak állapottér-reprezentációja	<b>10.13</b>
<b>7</b>	<b>Zárthelyi dolgozat</b> (állapottér-repr., Javában programozva is, keresőeljárások, minimax, alfabéta)	Egy teljes példa GUI-val a beadandóhoz való segítségként	<b>10.20</b>
<b>8</b>	Logika és ismeretreprezentáció	Beadandó 2. fázis: GUI egy állapotra, véletlenpálya-generálás	<b>10.27</b>
<b>9</b>	Elsőrendű logika és ismeretreprezentáció, következtetés	Beadandó 3. fázis: léptetés és operátorok megvalósítása	<b>11.03</b>
<b>10</b>	Automatikus következtetés, logikai programozás, Prolog, kényszerfeltételes logikai programozás	Beadandó 4. fázis: mentés, betöltés	<b>11.10</b>
<b>11</b>	Szakértői és döntéstámogatási rendszerek.	Logikai feladatok megoldása – tutorial.	<b>11.17</b>

Ismeretprezentációs  
módszerek. Szakértői  
rendszerek és logikai  
programozás.

<b>12</b>	Bizonytalan ismeretek ábrázolása, pl. Bayes- hálók. Tanuló algoritmusok: ID3. Az adatbányászat alapjai.	Szakértői rendszerekkel kapcsolatos gyakorlati feladatok	<b>11.24</b>
<b>13</b>	<b>Zárthelyi dolgozat</b> (logika, logikai programozás, Bayes- hálók, ID3-algoritmus)	Beadandó-bemutató és védés	<b>12.1</b>
<b>14</b>	<b>Ismétlés</b>	Ismétlés	<b>12.8</b>

Oktatási segédanyag:

Az előadáson bemutatott prezentációk megtalálhatók a <http://moodle.nyf.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)**