

BPI1235L (BPI1239L, BMI1602L, PMB1212L, PMB1218L) Mesterséges intelligencia

Tantárgyi tematika és félévi követelményrendszer

kontakt óraszám (elm.+gyak.): 9+9

Előfeltétel (tantárgyi kód) BPI1204 (programozás 2)

Oktató: Vályi Sándor (PhD)

- *elektronikus levelezőcím: valyi.sandor@nye.hu, 2 munkanapi reakcióidő, nem azonnali ügyfélszolgálat!*
- *A kurzushoz tartozik: classroom.google.com- Felvételi kulcs: os43fma itt fogunk gyakorló feladatokat megoldani interaktívan és az előadási anyagok is ide helyeztetnek*

Félévi követelmény: vizsga (évközi alapján lehetséges jegymegajánlás is)

Évközi követelmény, vizsgára bocsájtás feltétele:

beadandó program készítése és védeése [40 p]

2 darab dolgozat [45 p + 15 p]

A minimum követelmény 30 p elérése, ezzel lehet menni szóbelileg vizsgázni. 40 ponttól elégséges jegymegajánlás, 60-tól közepes, 75-től jó, 90-től jeles.

Az évközi követelményeket egyszer lehet csak ismételni várhatóan június elején, ezek alapján már nem kötelező a jegymegajánlás az oktató számára.

A vizsga: szóbeli. Itt a jegymegajánlás és az évközi pontok már nem számítanak, minden jegyet lehet szerezni 1..5 intervallumban.

Féléves tematika:

A félév 3 alkalomból áll.

1. 2024. március 1 8:00–15:45, 4 x 15 percz szünettel

2. 2024. ápr 12 8:00–13:00, 2x15 perc szünettel

Ez még nincs az órarendben: lesz még egy ZH-írási alkalom és egy beadandóvédési alkalom, alkalmasint egymás után egy napon, 3x45 perc időtartamban. Ennek ismétlése már a vizsgaidőszakban, még a záróvizsgás határidő előtt.

Az órák programja (tematika):

Alkalom	Előadáson	Gyakorlaton
1	Bevezetés, intelligens ágensek, generatív	Ágensek tulajdonságai; Egyszerű ágensek programozása,

	mestint használata programozásban	programozás és tesztelés chatGPT aktuális ingyenes verziójával
1	Problémák állapottér-reprezentáci ója	Példák állapottér-reprezentációra, Java keretrendszer állapottér-reprezentált problémák számára
1	Problémamegoldás, mint útkeresés	Mélységi, szélességi és egyéb nem informált keresők.
1	Informált keresési eljárások	Informált keresők
1	2-személyes játékok, Java keretrendszer a 2-személyes játékokhoz. Példa a beadandó 0-1-2. produktumára	Minimax algoritmus, alfa-béta vágás.
2	Adatbázist nem építő keresési eljárások. Genetikus algoritmusok, backtrack eljárás. Kényszerfeltételek az állapotokon.	Beadandó-témaválasztás.Beadan dó 0. feladat.
2	ZH-előkészítő	
1	Beadandó példaprogram bemutatása	
2	Nulladrendű és elsőrendű logika és ismeretreprezentáció, következtetés, automatikus következtetés: rezolúció ítéletlogikára	Ismeretreprezentáció Tarski világában. Logikai feladatok megoldása -- tutorial.
2	Logikai programozás, Prolog, kényszerfeltételes logikai programozás. Szakértői és döntéstámogatási rendszerek. Ismeretreprezentációs	Prolog gyakorlatok.

	módszerek. Szakértői rendszerek és logikai programozás.	
2	Bizonytalan ismeretek ábrázolása, pl. Bayes-hálók. Tanuló algoritmusok, Döntési fák.	Gyakorlatok a Bayes-hálókról, ID3 algoritmusról, döntési fákról
2	Neuronhálók, deep learning a szövegfeldolgozásban	Beadandó konzultáció.

Értékelési rendszer (évközi és vizsga)

-- évközi 1. követelmény: 40 pont **beadandó**, ami a későbbiekben részletesen le van írva.

Ennek benyújtása messze nem elég: védése a gyakorlatokon, ez, ami számít.

A beadandóválasztás az ötödik hétig a moodle-n. A beadandó köztes állapotainak bemutatása szintén, a gyakorlatokon. Benyújtási határidő a táblázatban. Minimum pontszám 24, anélkül nem lehet vizsgázni.

<i>Fázis</i>	<i>Leírás</i>	<i>Alkalom</i>	<i>Pontszám</i>
0. fázis	a feladat szöveges leírásának feltöltése, ez a belépő	2	0
1. fázis	a választott témák problémáinak állapottér-reprezentációja "papíron", állapotok, operátorok leírása egy dokumentumban. A dokumentáció tartalmazza az állapottér (játékfa) egy 10-12 állapotú részének ábráját, ahol a lehetséges operátorok is elmagyarázhatók, valamint azt, hogy milyen kereső lesz leprogramozva.	<i>védezen</i>	15
2. fázis	állapottér-repr. Javában programozva is, keresőalgoritmus beállítása, működik a	<i>védezen</i>	25

	keresés, tesztek bizonyítják, hogy jól működik. A program az állapotterében való keresés egyes lépéseit is írja ki, ne csak a megoldást.		
--	--	--	--

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 működnie.

A problémamegoldás/játék inicializálható legyen egy kezdőállással, amit input fájlból olvasunk föl. 2 személyes játék esetén az egyik ellenfél lépéseit humán vezesse, a másik fél lépéseit a minimax lépésajánló, legalább 8 féllépés mélységig.

Előny, ha a program grafikus felhasználói felülettel rendelkezik, pályaszerkesztési funkcióval (a kezdőállás grafikus megszerkesztése). Előny, ha annyi távoli adatbázis-használatot tartalmaz, hogy elmenti az adott felhasználó nevét, a megszerkesztett generált pályát egy megadott névvel, a játszmat és az eredményt, vagy relációs adatbázisban, vagy XML-adatként. Mondani sem kellene, vissza is kell tudnia tölteni a mentésből a név alapján kikeresve a több elementett állás közül.

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.

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, ő jeles megajánlott jegyet kap, s sejtetően mehet vele országos TDK-ra is.
- aki működő ágenszt készít reális (amely problémákat szokták mellékelni a PC-s sokobanhoz) Sokoban feladatok megoldására, jeles megajánlott jegyet kap.
- aki olyan webhelyet készít, pl. Java web service, Spring MVC (ez a preferált!) technológiára, amelyre a regisztrált userek felvihetnek Sokoban állásértékelő 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.

- más, saját ötletes, neuronhálós vagy más gépi tanulást tartalmazó feladatmegoldás, ami példaként bemutatatható a csoporttársaknak. legyen hozzá prezentáció is, kód is, a kód és a prezentáció open source legyen.

Az vizsgajegy kialakításának módja:

A teljesítménnyel pozitív korrelációban, a pontszámok alapján..

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>