

תרגיל בית 3

מבוא

מטרת התרגיל היא להתנסות בפתרון בעיית wumpus בתנאי אי-דטרמיניזם של הסביבה.

בדומה לתרגילים קודמים עליכם להביא את הגיבורים לתאים שבהם נמצא זהב. כמו כן, בדומה לתרגיל ראשון ובשונה מתרגיל 2 יש לכם ידע מלא על העולם. בשונה מתרגילים קודמים תוצאות של הפעולות מבוצעות אינן דטרמיניסטיות.

הבעיה

כקלט ראשוני תקבלו מפה מלאה (מטריצה בגודל $M \times N$) ועליה כל האובייקטים. בדומה לתרגיל 2, לאחר קבלת קלט עליכם להחליט על פעולה הבאה בפונקציה `get_next_action`. לאחר מכן הפעולה מבוצעת, ומפת העולם אחרי הפעלת פעולה נכנסת כקלט ל-`get_next_action` הבא. התהליך ממשיך עד סיום תקציב התורות.

הפעולות המותרות הן פעולות זהות לתרגילי בית 1 ו-2, בתוספת פעולה מיוחדת `reset`. בנוסף, האפקט של פעולות מסוג `move` אינו דטרמיניסטי, ויכול להביא את הגיבור למקום אחר ממה שתוכנן. פרטים מדוייקים ניתן למצוא בסעיף "פעולות".

בתרגיל זה המטרה היא לא הגעה לזהב, אלא איסוף כמות מרבית של נקודות בכמות פעולות קבועה. פרטים מדוייקים נמצאים בסעיף "נקודות".

תיאור משימה

לתרגיל זה מצורף קוד אשר משנה את המצב העולם אחרי כל פעולה ומעביר את המצב כקלט למחלקת `WumpusController` שעליכם לממש. במחלקה הזאת יש לממש פונקציה `get_next_action` אשר מקבלת מפה, ומחזירה את אחת מהפעולות החוקיות. התהליך נמשך למספר קבוע של תורות, ואז המשחק נגמר.

מספר התורות ייחודי לכל קלט, ומועבר ל-`constructor` כארגומנט. מספר התורות שנשאר מועבר ל-`get_next_action`.

האובייקטים על המפה זהים תרגיל בית 2, ומתנהגים באותה צורה.

פעולות

כל הפעולות שהיו חוקיות בתרגיל 1 נשארות חוקיות.

- פעולות מסוג `shoot` פועלות בדיוק באותה צורה כמו בתרגיל 1.
- פעולות מסוג `move` מזיזות את הגיבור למקום המתוכנן בהסתברות 0.7, הצידה בהסתברות 0.1 לכל צד, ובהסתברות 0.1 מעבירות את הגיבור עוד תא לכיוון התנוע. לדוגמה זאת תהיה התפלגות של תוצאות אפשריות אחרי הפעלת ("`move`", 11, "`R`"):

	0.1	
11 (גיבור)	0.7	0.1
	0.1	

במקרה ויש קיר או גיבור בתא שמנסים לעבור אליו, התזוזה תשאיר את הגיבור במקום בהסתברות 1.

במקרה ויש קיר או גיבור בתא שאפשר להגיע אליו בהסתברות 0.1, ההסתברות עוברת לתא היעד. למשל זאת תהיה התפלגות של תוצאות אפשריות אחרי הפעלת ("move", 11, "R") אם יש קיר ליד:

	קיר	
11 (גיבור)	0.8	0.1
	0.1	

אם התזוזה הייתה לכיוון הקיר או גיבור אחר מלכתחילה – אין שינוי במצב המשחק (בהסתברות 1)

- פעולה נוספת שביכולתכם לעשות היא reset – "reset" שמשמעותה היא החזרת המפה למצבה המקורי. הפעולה חוקית בכל מצב, גם כאשר אין גיבורים על המפה.

ניקוד

כמו שהוזכר מקודם, היעד שלכם בתרגיל זה הוא קבלת מספר מקסימלי של נקודות. הנקודות ניתנות בהתאם לחוקיות הבאה:

- כל פעם שהגיבור דורך על הזהב – הזהב נעלם ואתם מקבלים 10 נקודות
- כל ירי (הפעלת פעולה shoot) מוריד נקודה אחת
- כל הפעלת reset מורידה 15 נקודות
- כל שאר הפעולות והמצבים לא משפיעים על הניקוד

חבילות מותרות לשימוש

זהות לתרגיל 2

הסבר קצר על קוד מצורף

הקוד מורכב מ-5 קבצי קוד

- ex3.py – הקובץ שבו נעשית העבודה העיקרית שלכם, הקובץ היחיד שעליכם לשנות ולהגיש.
- checker.py – קובץ המכיל את הקוד שממדל את התנהגות העולם כתוצאה מפעולות שאתם מבצעים
- inputs.py – קובץ המכיל מפות שונות לצורך בדיקה עצמית
- utils.py – קובץ עזר
- mdp.py – קובץ המכיל פונקציות עזר בנוגע ל-MDPs.

בדיקת התרגיל

התרגיל ייבדק באופן אוטומטי, ולכן חשוב להקפיד על שמות מדויקים של קבצים, מחלקות ופעולות. הבדיקה תבצע על ידי הרצת קובץ checker.py

זמנים שהקוד חייב לעמוד בהם - ה constructor חייב להסתיים ב-60 שניות, get_next_action חייב להסתיים ב-5 שניות.

כל קלט ירוץ מספר רב של פעמים על מנת לנטרל את השונות, וכל ציון ייקבע על סמך ממוצע של תוצאות הריצות.

נא להוריד את כל ההדפסות לפני הגשת התרגיל!

הגשה, בדיקת תרגיל וציונים

- ההגשה ביחידים או בזוגות בלבד
- מספיק שאחד מבני הזוג יגיש את התרגיל, אין צורך ששני בני הזוג יגישו
- את ת.ז. של המגישים יש לרשום במשתנה ids בקובץ ex3.py בתור strings ברשימה קיימת
- יש להגיש רק את הקובץ ex3.py, אין להגיש קבצי עזר. אין להגיש בפורמט אחר (zip, rar וכו')
- 80% מהציון יינתן על עמידה בדרישות בסיס, ו 20% על בסיס תחרות בין המגישים.
- דרישות הבסיס - להגיע לתוצאה חיובית ממש בכל קלט כך ששום מימד של מטריצת הקלט לא עולה על 8
- ציון על תחרות יקבע בצורה פרופורציונלית לניקוד שהסוכן צבר לאורך הריצה