

מבוא לבינה מלאכותית

סמסטר חורף תשפ"ו

מטלה 2

תאריך הגשה: 23:55 10.12.25

הנחיות

- שאלות בנושא מטלה זו יש לשאול דרך המודל, בפורום "מטלה 2".
- הוראות להגשת המטלה מופיעים בסוף מסמך זה.
- הקבצים הנדרשים להרצת הקוד הינם:
 - minimax_isoKnight.py
 - player_agent.py
 - heuristics.py
 - alpha_beta_isoKnight.py
 - heuristic_alpha_beta_isoKnight.py
 - game_state.py
 - game_engine.py
- העבודה להגשה בזוגות בלבד אלא אם ניתן אישור מיוחד להגיש אחרת.
- לפני שניגשים למשם את המטלה מומלץ לעיין רבודת בהסברים וכן בקוד הקיימ.
- פתרון המטלה שתגלו יידק מול שאר ההצעות על ידי תוכנת העתקות.
- **מי שימצא כי העתיק יכול בקורס וכן יועבר לוועדת משמעת אוניברסיטאית.**
- הפרויקט נכתב ויבדק בשפת התוכנות Python.
- יש להתקן את החבילה `osunut`.
- מסמך זה בניו באופן הבא: תיאור המטלה, בעיתת משחק שני שחknim, מרחב הבעיה, שאלות המטלה, הסבר על המימוש הקיימ והסביר על הגשת המטלה.

תיאור המטלה

נתון משחק שני שחknim בשם IsoKnight. בהינתן לוח משחק $n \times m$, לכל שחkn יש דמות המתחילה במקומות מסוימים על הלוח. כל שחkn בתווך יכול להזיז את הדמות שלו בדיקן כמו פרש בשחמט (הסבר בהמשך). הדמות יכולה לזרז לכל משבצת בכיוון מסוים אם שלושת התנאים הבאים מתקיימים:

1. לא ניתן להגיע למשבצת שכבר ביקרו בה
2. ניתן לדלג מעל משבצות שביקרו בהן
3. ניתן לדלג מעל לדמות השחקן השני

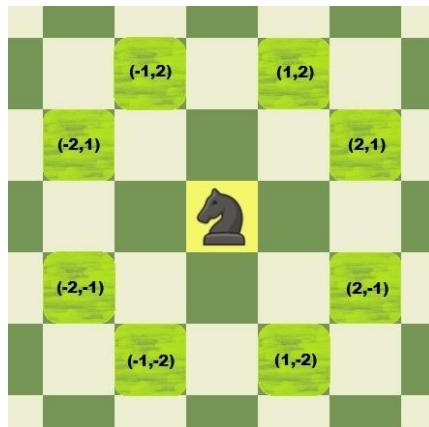
כיצד פרש זו? – לפרש יש שני צעדים אפשריים:

שני צעדים באחד מרבעת הכוונים (למעלה, למטה, ימינה או שמאליה) ולאחר מכן צעד אחד הצד.

צעד אחד באחד מרבעת הכוונים (למעלה, למטה, ימינה או שמאליה) ולאחר מכן שני צעדים הצד.

תמונה הממחישה את כל הצעדים האפשריים בעמוד הבא.

הפרש יכול לדלג מעל כל המשבצות פרט לצעד האחרון (התא אליו הוא עובר).



בעיה משחק שני שחקנים

בהתנאי מיקום התחלתי של שני השחקנים, עליוכם למשת אלגוריתמים שיחשבו את ערך ה- `maximin` של המשחק.

מרחב הבעיה

בכל שלב במשחק, הדמות נמצאת באחד התאים במטריצה, כאשר המטרה שלה היא שלדיםות השניה יגמרו המהלים לפניה.

דוגמה ללוח משחק:



שאלות המטלה

- עליכם למשת אלגוריתם `alphabeta`. שם כר האלגוריתם `minimax` ו-`maximin` ממומשים בקובץ `py`, `minimax_isoKnight`, ואילו את אלגוריתם `alphabeta` יש לשנות ולהתאים בקובץ `py`. האלגוריתם המומש כעת בקובץ `alpha_beta_isoKnight.py` הינו גם `minimax` ו-`maximin`, ויש לשנות את הפונקציות בקובץ כר שהאלגוריתם יימומן יהיה (`alpha_beta_pruning`).
שים לב, כאשר מוצאים גיזום יש להחזיר את `v` וגם `None`, מכיוון שתוכנית זו שומרת את הצעד שיש לבצע עבור השחקן הפורט את עץ המשחק (כלומר, `v`, `None`). ניתן לשנות את חתימות הפונקציה בשביל נכונות האלגוריתם `alphabeta`.

- עליכם למש את אלגוריתם heuristic_alpha_beta_isoKnight.py גם בקובץ alphabeta.py (אוותם שינויים כמו בסעיף א'). בנוסף, כדי להתמודד עם עצי משחק גדולים יותר- עליכם למש יוריסטיקה עבור קודקודים שאין מציינים סיום משחק. השלימו את הפונקציה base_heuristic בקובץ heuristics.py כך שתקבל מצב של משחק ותחזיר את הפרש בין כמהות הצעדים החוקיים של השחקן הראשון לבין הצעדים החוקיים של השחקן השני. כמובן, עליכם תמיד להחזיר moves_1 - moves_2 (לא משנה תור מי לשחק בצעד הנוכחי).
 - הינכם יכולים למש יוריסטיקה מתקדמת בפונקציה advanced_heuristic.py. היוריסטיקה הטובה ביותר תזכה את המגישים ב 2 נקודות בונוס לציון הסופי. כדי להחיליט על היוריסטיקה הטובה ביותר ביותר, נעורו טורניר בין קבוצות המגישים כך שהסטרטגיה של השחקן תשתמש ב advanced_heuristic על מנת להחליט על כל אחד מצעדי המשחק. הקבוצה שתנצח בטורניר תקבל את נקודות הבונוס.
- שימוש לב:** כאשר אתם מרים את אלגוריתם minimax אתם תוכלו לחשב משחקים לוח בגודל 5x5 (ולפעמים 6). כדי לנסוט לשחק על לוח בגודל 6x6 עליכם למש את pruning.alphabeta בשביל לוחות גדולים יותר לצריכו להשתמש ביוריסטיקה ולהגביל את עומק החיפוש.

הגשת המטלה

- יש להגיש **שלושה קבצים בלבד** לאתר המודול בדומה להגשת המטלה הקודמת. הקבצים הנדרשים:
 - heuristics.py
 - alpha_beta_isoKnight.py
 - heuristic_alpha_beta_isoKnight.py

- הגשת המטלות תבצע ישירות מול מערכת המודול בצורה אלקטרוני.
- שימוש לב:** ישנה אפשרות להגיש את המטלה התכניתית מספר פעמים ובכל הגשה לקבלת חיווי, כמובן תקבלו באופן מיידי את הציון לביצוע התרגיל.
- ניתן לראות את הפידבק להרצאה (קומפיילציה, מספר טיסטים בעברן, שגיאות זמן ריצה וכו'...).

לאחר סיום הרצאה יתקבלו התוצאות.