מבוא לבינה מלאכותית – תרגיל 2

מאיר כהן ושירה נגבי

חלק א' – הבנת המשחק (20 נק')

בשלב ראשון, נרצה לקבל הבנה בסיסית להגדרת מרחב המצבים מעליו נפעל. שימו לב כי בחלק זה (חלק א') **אין צורך לכתוב קוד**.

1. פתחו את הקובץ $checkers/board.py$ והסתכלו על המחלקה $GameState$. מהו המבנה של כל מצב ($state$) במרחב המצבים של המשחק?
2. במחלקה $GameState$, הפונקציה $get\_possible\_moves$ מחזירה את כל המהלכים (אופרטורים) שניתן לבצע מהמצב הנוכחי. כל מהלך מוחזר כאובייקט $GameMove$ המוגדר בקובץ $checkers/moves.py$. איזה מידע נחוץ כדי לייצג כל אופרטור? איך ניתן להבדיל מעל מבנה זה בין תנועה רגילה ($movement$) לפעולות קפיצה ($capture$)?

סעיף א

המצב של כל מצב במשחק:

① לכל מצב המשחק מספיק רק ג'יג שלו כלי: אבן, מלך, אבן-מלך, מלך-מלך
או שבכלל רק איזו אבן שלו בלי.

② רשימות של קואורדינטות אפשריות: תנועות לכיוון או קואורדינטה הנוכחית.

③ מי מהלכים מהמצבה האחרונה.

④ תור של השחקן הנוכחי.

2 - א. המידע הדרוש בכדי לייצג אופרטור:
- מבט שחקן.
- מיקומו ומיקום שאר החלקים הקבועים.
- סוג הקפיצה.

ומקבלים רשימה של possible moves מהפונקציה.

ב. במידה ויש פעולת קפיצה, הפונקציה תחזיר אך ורק את אפשרויות הקפיצה (capture) ומסיים.
במידה ואין הפונקציה תחזיר קואורדינטה הנוכחית (movement).

חלק ב' – הבנת השחקן הפשוט (12 נק')

קראו את הקוד של השחקן הפשוט המסופק כדוגמא ב *players/simple_player*. שימו לב כי בחלק זה (חלק ב') **אין צורך לכתוב קוד**.

3. מהי הגישה (הנאיבית) בה נוקט השחקן לשם חלוקת הזמן בין כל k מהלכים?
4. ציינו חסרון אפשרי של גישה זו והסבירו כיצד שחקן חכם יותר יוכל להתגבר עליו.
5. מהי הפונקציה היוריסטית בה משתמש השחקן? הסבירו את המוטיבציה ליוריסטיקה זו.

.3

```
# We are simply providing (remaining time / remaining turns) for each turn in round.
# Taking a spare time of 0.05 seconds.
self.turns_remaining_in_round = self.k
self.time_remaining_in_round = self.time_per_k_turns
self.time_for_current_move = self.time_remaining_in_round / self.turns_remaining_in_round - 0.05
```

השחקן מחלק את הזמן עבור כל תור כך: בהינתן סך כל הזמן הנותר לו בסיבוב הנוכחי וכן מספר התורות שנותרו, הוא בוחר את הזמן הנותר חלקי מספר התורות (עם מרווח ביטחון של חמש מאיות השנייה). זאת ללא קשר למצב הלוח.

4. חסרון אפשרי: חישוב זה מתעלם ממספר האפשרויות השונות שיש לבדוק או ממידת הקריטיות של ההחלטה. במצב שבו יש מעט שחקנים, למשל, ייתכן שיהיו מעט אפשרויות, ולכן לא יידרש זמן חישוב ארוך במיוחד. לעומת זאת, כדאי כן להשקיע זמן חישוב במצבים שקרובים לסיום המשחק, כלומר, כשיש איום קרוב, נרצה להימנע ממנו מראש ככל האפשר. שחקן חכם יותר יוכל להקצות זמן קצר יותר למקרים בהם יש מעט אפשרויות על הלוח או כשיש אפשרויות עדיפות כמו יצירה של מלכה, וזמן ארוך יותר למקרים בהם יש סכנה להפסד קרוב ושכדאי לחשב את האפשרויות השונות לעומק לפני שמקבלים החלטה.

.5

```
my_u = ((PAWN_WEIGHT * piece_counts[PAWN_COLOR[self.color]]) +
        (KING_WEIGHT * piece_counts[KING_COLOR[self.color]]))
op_u = ((PAWN_WEIGHT * piece_counts[PAWN_COLOR[opponent_color]]) +
        (KING_WEIGHT * piece_counts[KING_COLOR[opponent_color]]))
if my_u == 0:
    # I have no tools left
    return -INFINITY
elif op_u == 0:
    # The opponent has no tools left
    return INFINITY
else:
    return my_u - op_u
```

השחקן משתמש בפונקציה היוריסטית utility, שבעזרתה הוא מעריך את מצבי המשחק השונים. הוא נותן ציון לכל מצב אפשרי כך: עבור כל אחד מהשחקנים, עצמו והיריב, הוא נותן ניקוד שנקבע על-פי מס' החיילים בצבע שלו כפול הערך הקבוע של חייל, ועוד מס' המלכים בצבע שלו כפול הערך הקבוע של מלך. הציון של המצב הוא הניקוד של עצמו עבור אותו מצב פחות זה של היריב.
המוטיבציה ליוריסטיקה זו היא ההנחה שככל שיש לשחקן מסוים יותר כלים על הלוח, יש לו יותר סיכוי לנצח. זאת כי השחקן הראשון שנגמרים לו הכלים, מפסיד. כך מי שמאבד כלים מתקרב אל ההפסד. כמו כן, מלכים הם חזקים יותר מחיילים פשוטים מהסיבה שיש להם הרבה יותר אפשרויות לתנועה, ולכן הערך שלהם ככלים גבוה יותר מזה של חיילים פשוטים.

חלק ג' – שיפור השחקן (24 נק')

בחלק זה של התרגיל תדרשו לממש שלושה שחקנים. על כל שחקן להיות ממומש בתיקייה נפרדת על פי שמו. מחלקת השחקן תקרא $Player$ ועליה לרשת מהמחלקה $AbstractPlayer$ (או מהמחלקה $Player$ של $simple\_player$).

6. <u>שיפור היוריסטיקה</u>:
   I. הגדירו באופן מפורש יוריסטיקה משלכם להערכת מצבי המשחק. בפרט, יש לספק נוסחה עבור כל פרמטר של המשחק הנלקח בחשבון לשם חישוב היוריסטיקה.
   II. הסבירו את המוטיבציה להגדרה זו (ולכל פרמטר המופיע בה). מדוע אתם צופים שהיא תשפר את ביצועי השחקן ביחס ליוריסטיקה של $simple\_player$?
   III. ממשו שחקן בשם $better\_h\_player$ הנעזר ביוריסטיקה שהגדרתם.

6. היוריסטיקה שבחרנו להערכת מצבי המשחק מתחשבת, בנוסף לחישוב התועלת של השחקן הפשוט, בפרמטרים הבאים, עליהם היא מוסיפה ניקוד למצבים:

- מספר הכלים שנשארו בשורה האחרונה. מוטיבציה: ככל שהשורה האחרונה מלאה יותר, כך יהיה קשה יותר ליריב ליצור מלכים. חישוב: הגדרנו קבוע לבונוס על הימצאות כלי בשורה האחורית, וחישבנו (מספר הכלים שבשורה האחורית x קבוע הבונוס) + (קבוע הבונוס במקרה בו השורה מלאה לגמרי או קבוע בונוס קטן יותר במקרה בו השורה מלאה פרט למקום אחד). מחסרים את הערך שהתקבל עבור כלי היריב מהסכום שהתקבל עבור כלי השחקן (power_a – power_b).
- קרבת החיילים לשורה הקדמית ביותר, כלומר, הרחוקה ביותר מהצד שבו התחיל השחקן. מוטיבציה: ככל שחיילים יהיו קרובים יותר לשורה זו, כך יוכלו במהירות יותר להפוך למלכים ולתת יתרון לשחקן. חישוב: |מספר השורה הקדמית ביותר – מספר השורה בה נמצא החייל|. מחסרים את הסכום שהתקבל עבור חיילי השחקן מהסכום שהתקבל עבור חיילי היריב. (promoteA – promoteB) –.
- מספר הכלים בדפנות הצדיות של הלוח. מוטיבציה: כלי שנמצא בצד הלוח לא יכול להיאכל על-ידי היריב. חישוב: נקודה לכל כלי של השחקן שנמצא בטור הימני ביותר בלוח או בטור השמאלי ביותר בלוח. מחסרים את הערך שהתקבל עבור כלי היריב מהסכום שהתקבל עבור כלי השחקן (my_score_row – op_score_row).

לבסוף מתבצע השקלול של כל הפרמטרים ביחד, וזוהי ההערכה של כדאיות המעבר למצב מסוים בעץ המצבים.

```
return (power_a - power_b) + (my_score_row - op_score_row) - (promoteA - promoteB) + (my_sides - op_sides)
```

את השינויים ביצענו בפונקציה utility.

כעת נרצה לממש שיפור לאלגוריתם $Minmax - \alpha\beta$ בכדי לשפר את ביצועי השחקן שלנו. את מימוש השחקן המשופר (ללא היוריסטיקה משאלה 6) יש לשמור בשם $improved\_player$.

7. <u>ניהול זמנים</u>:
   I. הגדירו שיטה אינטליגנטית לחלוקת הזמן של השחקן בין כל k מהלכים. שיטה נאיבית ניתן, כאמור, למצוא בשחקן המסופק לכם. יש להציג נוסחה כללית במקרה הצורך, ונוסחה עבור כל פרמטר של המשחק הנלקח בחשבון לשם חישוב זה.
   II. הסבירו את המוטיבציה מאחורי השיטה שמימשתם. באילו מצבים השיטה משקיעה פחות זמן חיפוש, ובאילו יותר? מדוע לדעתכם פעולה זו תהיה משמעותית לשיפור השחקן הפשוט?
   III. ממשו שחקן עם מדיניות ניהול הזמנים שהגדרתם.

7. השיטה שבחרנו לחלוקת הזמן של השחקן בין k מהלכים מבוססת על כמות המהלכים האפשריים שיש לשחקן בתור הנוכחי, בנוסף להתחשבות בזמן הכולל שנותר לסיבוב הנוכחי, ובמספר התורות שנותרו. למעשה שיפרנו את השיטה הנאיבית של השחקן הפשוט, שלוקחת בחשבון רק את הזמן הכולל הנותר ואת מספר התורות הנותרים.
הנוסחה היא:

זמן התור = סך הזמן שנותר \ (מספר המהלכים האפשריים x מספר התורות שנותרו)
(פחות מרווח הביטחון).

```
def get_move(self, game_state, possible_moves):

    # Estimate the time needed for the current turn considering the number of possible moves
    estimation = len(possible_moves) * self.turns_remaining_in_round
    self.clock = time.process_time()
    self.time_for_current_move = self.time_remaining_in_round / estimation - 0.05
```

המוטיבציה מאחורי שיטה זו היא שהמצבים שבהם כדאי להקדיש יותר זמן לבחינת האפשרויות, הם המצבים בהם השחקן קרוב להפסד. מצבים אלה יתקבלו דווקא כשיש מעט מהלכים אפשריים שונים לשחקן, ונסביר: בתחילת המשחק יש לשחקן הרבה כלים, ולכן יש הרבה אפשרויות למהלכים, אבל בשלבים אלה אין סכנה של הפסד בטווח הקצר, ולכן לא נרצה לבזבז הרבה זמן. עם התקדמות המשחק יש לשחקן פחות כלים, ואם הם חיילים פשוטים, יש לו מעט אפשרויות לתנועה. בו בזמן יש לו גם סכנה להפסיד בקרוב, ולכן אלה תורות קריטיים שכדאי להשקיע בהם זמן לבחירת המהלך הטוב ביותר. כשחייל הופך למלך, מספר האפשרויות למהלכים גדל, כי למלך יש יותר אפשרויות תנועה על הלוח, ובנוסף השחקן מתחזק, ולכן הסכנה להפסד קרוב קטנה. עם זאת, ספירה של כמות החיילים הפשוטים ושל כמות המלכים עשויה לצרוך זמן יקר בפני עצמה, ולכן השתמשנו ישירות במספר המהלכים האפשריים. מספר זה מתקבל פשוט מאורך רשימת המהלכים האפשריים המועברת לפונקציה get_move בפרמטר possible_moves. בסך הכל, שחקן המחלק את זמנו בשיטה שבחרנו, ייכנס לעומק עץ המצבים בתורות הקריטיים יותר, ולא יתעכב יותר מדי בתורות האחרים, וכך יתנהל ביעילות ויגדיל את סיכויו לנצח.

10. נרצה להשוות את ביצועי השחקנים על פני מגבלות הזמן השונות:

I. הציגו גרף המתאר עבור כל אחת ממגבלות הזמנים את הניקוד הכולל של כל שחקן כפונקציה של מגבלת הזמן. הקפידו להשתמש בצבעים או בסוגי קו שונים עבור שחקנים שונים. לנוחיותכם מצורף גרף לדוגמא (*).

II. עבור גרף זה, צרפו גם את טבלת הנתונים שיצרו אותו.

III. השוו את ביצועי השחקנים עם היוריסטיקה הפשוטה (שחקנים (1) ו(3)) לאלו עם היוריסטיקה שהצעתם בשאלה 6 (שחקנים (2) ו(4)) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה בניקוד הכולל של כל אחד מהשחקנים. באילו מקרים היוריסטיקה שלכם מתגברת על היוריסטיקה הפשוטה? הציעו הסבר לממצאים הנ"ל.

IV. השוו את ביצועי השחקנים בעלי השיפור משאלה 7 (שחקנים (3) ו(4)) לאלו ללא שיפור זה (שחקנים (1) ו(2)) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה של כל אחד מהשחקנים. באילו מקרים השיפור שלכם משפר את ביצועי השחקן? הציעו הסבר לממצאים הנ"ל.

10. ניתוח התוצאות:
נשים לב שהגרף מתאר t שגדל מימין לשמאל.

	2	10	50
simple_player	3.5	4	3
better_h_player	4	2	3
improved_player	2	2.5	2
improved_better_h_player	2.5	3.5	4

נשווה את השחקנים המשתמשים ביוריסטיקה שלנו (better_h_player, better_h_improved_player) לאלה שלא (simple_player, improved_player):
השחקנים המשתמשים ביוריסטיקה גוברים באופן חלקי על השחקנים האחרים כאשר הערך של t קטן או בינוני אך מראים תוצאות טובות יותר כאשר הערך של t גדול. כלומר, בהינתן זמן רב יותר לכל תור, הם מצליחים להפעיל שיקול דעת חכם יותר ובכך לבחור במהלכים טובים יותר.

כעת נשווה את השחקנים המשתמשים בשיטת ניהול הזמן שלנו (improved_player, improved_better_h_player) לאלה שלא (simple_player, better_h_player): השחקנים עם שיטת ניהול הזמן שלנו אמנם מפסידים לשחקנים האחרים עבור ערך קטן של t, אך מראים תוצאות טובות יותר עבור ערך בינוני או גדול. כלומר, ייתכן שחלוקת הזמן עורכת בעצמה קצת יותר זמן מאשר חלוקת הזמן הנאיבית, אך הקרבה זו משתלמת עבור תורות ארוכים יותר.

בסך הכל נשים לב שהשחקן better_h_player משתפר ככל שיש יותר זמן לכל תור, כלומר, השילוב בין היוריסטיקה המשופרת שלנו לבין שיטת ניהול הזמנים שלנו מאפשר לשחקן הן להתעמק בעיקר בתורות החשובים והן למצוא בהם את המהלכים העדיפים בהינתן הזמן שניתן לו ובאמצעות דרך הערכת המצבים שהגדרנו, ולכן משפר את הביצועים לעומת השחקן הפשוט ומראה תוצאות טובות.