Exercise 3 – Checkers

Dorin Keshales – 313298424
Sagi Nachum – 034477588

חלק א' - הבנת המשחק

1. כל State במרחב המצבים של המשחק מכיל את לוח המשחק - לוח שגודלו 88 משבצות, כאשר המשבצות הן משבצות לבנות ומשבצות שחורות לסירוגין. כל משבצת מתוארת באמצעות tuple מהצורה (i,j), כאשר i הוא מספר השורה בלוח החל מ-0 ו-j הוא מספר העמודה בלוח החל מ-0, גם כן. כלי המשחק יכולים להימצא ולנוע רק על גבי המשבצות השחורות בלוח.
לוח המשחק מתעדכן עם כל פעולה המתבצעת מצד כל אחד מהשחקנים על מנת לאפשר את התקדמות המשחק מתור לתור.

בעבור כל משבצת שחורה בלוח המשחק, נשמר מידע שאומר האם המשבצת ריקה או אם כלי משחק מסוים נמצא בה ואם כן, באיזה כלי מדובר.

<u>הנוטציות לכך, הן:</u>
- אם על המשבצת נמצא חייל רגיל אדום, המשבצת תסומן עם 'r'.
- אם על המשבצת נמצא מלך אדום, המשבצת תסומן עם 'R'.
- אם על המשבצת נמצא חייל רגיל שחור, המשבצת תסומן עם 'b'.
- אם על המשבצת נמצא מלך שחור, המשבצת תסומן עם 'B'.
- אם המשבצת ריקה מכליי משחק (אף כלי אינו נמצא בה), המשבצת תסומן עם ' '.

בנוסף, כל State במרחב המצבים של המשחק מכיל את המידע של תור איזה שחקן לשחק כרגע - שחור/אדום ואת מס' התורות שעברו מאז הקפיצה האחרונה במשחק.

* בעת שאנחנו יוצרים אובייקט מסוג GameState הוא ישירות מאותחל עם לוח משחק המתאים למצב תחילת משחק - כלומר שלוש השורות הראשונות בלוח (במשבצות השחורות כמובן) מאותחלות עם כלי משחק אדומים ושלוש השורות האחרונות בלוח מאותחלות עם כלי משחק שחורים, כאשר באופן דיפולטיבי השחקן האדום מוגדר להתחיל את המשחק ומד התורות מאז הקפיצה האחרונה מאותחל עם 0.

2. המידע הנחוץ על מנת לייצג כל אופרטור אפשרי, הינו:

- איזה סוג שחקן מבצע את התנועה – אדום / שחור.
- משבצת המקור - המיקום הנוכחי של כלי המשחק על הלוח, לפני ביצוע התנועה.
- משבצת היעד - המשבצת אליה כלי המשחק יעבור בעקבות ביצוע התנועה. בעת קפיצה מרובה, משבצת היעד היא המשבצת אליה נגיע לאחר ביצוע הקפיצה האחרונה.
- רשימה של כלי המשחק (של היריב) מעליהם כלי המשחק שמבצע את התנועה קופץ בדרכו ממשבצת המקור למשבצת היעד (רלוונטי רק במקרה ומדובר בפעולת קפיצה).

באמצעות אותה רשימה מהסעיף האחרון אנחנו יכולים להבדיל בין תנועה רגילה לבין פעולת קפיצה - אם הרשימה ריקה הרי שהתנועה שבוצעה היא תנועה רגילה, שכן לא בוצעה קפיצה מעל אף כלי משחק של היריב ובאם הרשימה אינה ריקה, אז בוצעה לפחות פעולת קפיצה אחת (פעולת 'קפיצה מרובה' במידה ומס' האיברים ברשימה הוא > 1).
בנוסף, כמובן שניתן להבדיל בין תנועה רגילה לבין פעולת קפיצה גם לפי המרחק בין משבצת המקור למשבצת היעד – אם מדובר במשבצות סמוכות אז מדובר בתנועה רגילה. אחרת, מדובר בקפיצה.

חלק ב' - הבנת השחקן הפשוט

3. הגישה הנאיבית בה נוקט השחקן לשם חלוקת הזמן בין כל k המהלכים היא פשוט חלוקה שוויונית של הזמן בין k המהלכים, כאשר הוא משאיר זמן ספייר של 0.05 שניות מכל תור לצורך אדמיניסטרציה.

4. נניח כי עבור k מהלכים השחקן מקבל total_time זמן. על פי הגישה הנאיבית של simple_player לחלוקת הזמן, עבור כל מהלך מ-k המהלכים יוקצה total_time/k זמן כאשר 0.05 שניות מזמן זה מוקצה לאדמיניסטרציה.
השחקן משתמש בזמן היחסי שהוקצה לו פר מהלך בכדי להפעיל את אלגוריתם Minimax בשיטת iterative-deepening search ולהעמיק את החיפוש ככל שמתאפשר לו במסגרת זמן זה. כלומר, אנחנו מתחילים מ- limit = 1 לאחר מכן מעמיקים את החיפוש באמצעות limit = 2 וכך הלאה עד שמסתיים הזמן שהוקצה למהלך.

<u>בשיטה זו אנחנו רואים שני חסרונות עיקריים:</u>

a. פוטנציאל לבזבוז זמן מיותר בהעמקה מעבר לסף עומק מסוים, כאשר אין כל צורך בכך עבור המהלך הנוכחי. כלומר, בעבור מהלך מסוים יתכן והעמקה נוספת מעבר לסף עומק מסוים לא תביא לנו כל שיפור או תועלת, בעוד שבעבור המהלכים הבאים אולי כן נצטרך להעמיק יותר את החיפוש, אך לא נוכל לבצע זאת מפאת חוסר זמן, שכן ההעמקה שתידרש תזדקק אולי ליותר מה- total_time/k זמן המוקצה בגישה זו לכל מהלך.

b. בגישה הנוכחית אין לוגיקה חכמה המאפשרת זיהוי state 'מיוחד' או במילים אחרות state בעל פוטנציאל להשפיע על המהלך הבא של השחקן, אשר מצדיק העמקה מעבר למגבלת העומק הנוכחית. כלומר, במצב הנוכחי, השחקן סורק בכל פעם את העץ עד עומק limit, כאשר ערך ה-limit גדל ככל שמתאפשר במסגרת הזמן המוקצה למהלך, אך יתכן כי בעומק limit האחרון אליו ניתן להגיע במסגרת הזמן, קיים state ייחודי של הלוח שדווקא היינו רוצים להעמיק את החיפוש בו, שכן יכולה להיות לו השפעה על ההחלטה הסופית שלנו למהו המהלך הבא, מה גם שאולי ה-state הזה הוא בעל פוטנציאל להוות לנו סיכון בהמשך המשחק ולכן, חשוב לנו לזהות מצבים כאלו בכדי לשמר את היכולת שלנו לנצח במשחק. לשחקן הפשוט אין לוגיקה חכמה לזיהוי מצבים כאלה, מה שיפגע לו ביכולת לבחור מהלכים חכמים לאורך המשחק ואף עלול להביא להפסדו.

<u>לדעתנו, שחקן חכם היה נוקט בשני אמצעים על מנת להתגבר על חסרונות אלה:</u>

a. הוא היה קובע ערך threshold להעמקה ולאחר שהגיע אל הסף הזה היה מפעיל לוגיקה חכמה המחליטה האם יש צורך בהעמקה נוספת עבור המהלך הנוכחי או לא. במידה ונקבע כי אין צורך בהעמקה נוספת, הזמן שחסכנו ישמר למהלכים הבאים עבורם ייתכן שהלוגיקה החכמה תקבע שכן יש צורך בהעמקה נוספת. ובכל מקרה, במהלך ה-k, השחקן ינצל את כל שארית הזמן שנותר, שכן בתור הבא הקצאת הזמנים מתאפסת.

b. הוא היה מפעיל לוגיקה לזיהוי state ייחודי שהלוח נמצא בו, ומעמיק ב-state כזה את החיפוש גם אם הגיע לעומק limit האחרון אליו ניתן להגיע במסגרת הזמן,. לדוגמא: נניח והשחקן הגיע לצומת בעומק limit וה-state שבעומק זה מחייב מהלך קפיצה של אחד השחקנים, השחקן החכם יכול להחליט להמשיך להעמיק

בצומת הנוכחי על מנת לבחון את המשך המשחק, על אף שלבאורה ה-limit אמור היה להגביל אותו. באופן זה, השחקן החכם מרוויח מחקר עמוק יותר של מצבים מעניינים או אף כאלו בעלי פוטנציאל להוות סיכון בהמשך המשחק, ובכך משיג יתרון על פני השחקן הפשוט.

5. הפונקציה היוריסטית בה משתמש simple_player מחשבת מעין ניקוד לכל שחקן לפי סוג וכמות הכלים שנשארו לו על הלוח, כאשר המשקל של מלך הוא 1.5 והמשקל של חייל רגיל הינו 1. הפונקציה מחזירה את הפרש הניקוד בין השחקן ליריב כ-utility של ה-state הנוכחי.
המוטיבציה ליוריסטיקה הזו נובעת מהרצון לבצע בכל שלב את המהלך שיבטיח לשחקן את ההפרש האפשרי הגדול ביותר מהיריב לפי כמות וסוג הכלים שנותרו על הלוח. לפי הגיון זה, ככל שההפרש יהיה גדול יותר, כך הסיכויים של השחקן להגיע למצב ניצחון (מצב בו ליריב אין יותר כלים על הלוח) גדולים יותר ולהפך.

חלק ג' - שיפור השחקן

שיפור היוריסטיקה:

6. היוריסטיקה שהגדרנו לוקחת בחשבון מספר פרמטרים וממשקלת ביניהם, חלק מהפרמטרים נועדו על מנת לתת יתרון בשלבים מוקדמים של המשחק וחלק נועדו על מנת לתת יתרון בשלבים מאוחרים יותר.

להלן פירוט הפרמטרים בהם אנו מתחשבים במסגרת היוריסטיקה:

a. כמות הכלים על הלוח
בפרמטר זה אנו סופרים כמה כלים יש לכל שחקן על הלוח - חיילים רגילים ומלכים, ומחשבים את ההפרש בין מספר הכלים שלנו למספר הכלים של היריב, כאשר ערך חיובי משקף יתרון (מבחינת כמות כלים על הלוח) לשחקן שלנו וערך שלילי משקף יתרון ליריב. הערך עצמו (חיובי או שלילי) משקף את עוצמת היתרון.
נוסחא: `piece_difference = my_pieces – opponent_pieces`
המוטיבציה לשימוש בפרמטר זה היא טריוויאלית, שכן ככל שהפרש הכלים גדול יותר לטובת השחקן שלנו, כך מצבו במשחק טוב יותר ולהיפך.

b. כמות המלכים על הלוח
בשונה מהפרמטר הקודם, כאן אנו סופרים רק את מספר המלכים שיש לכל שחקן על הלוח, ומחשבים את ההפרש בין מספר המלכים שלנו למספר המלכים של היריב, כאשר ערך חיובי משקף יתרון לשחקן שלנו וערך שלילי משקף יתרון ליריב. הערך עצמו (חיובי או שלילי) משקף את עוצמת היתרון.
נוסחא: `king_difference = my_kings – opponent_kings`
המוטיבציה לשימוש בפרמטר זה נועדה לתת boost נוסף בחישוב ה-utility לשחקן שיש לו יותר מלכים, מאחר שמלך נחשב לכלי "חזק" יותר על הלוח ובאופן כללי ליתרון במשחק.

c. מיקומים אסטרטגיים על הלוח
פרמטר זה נועד לתת משקל להצבת כלי השחקן במיקומים מסוימים על הלוח ובכך 'לעודד' אותו לתפוס מיקומים אסטרטגיים על הלוח ולהמשיך להחזיק בהם. בנוסף, כפי שיפורט בסעיף v מטה, הפרמטר מתחשב גם במצב הלוח לאורך המשחק, כך למשל אם מספר הכלים על הלוח הוא דל אנו ננקוט באסטרטגיה שונה לחישוב הצעד הבא של השחקן.

הפרמטר מכיל בתוכו מספר לוגיקות, כדלהלן:

i. חייל רגיל הנמצא במיקום 'רגיל' (שאינו מיקום אסטרטגי), יקבל 3 נקודות על הימצאו במיקום זה.

ii. חייל רגיל הנמצא בשולי הלוח (בטור 0 או 7), יקבל 3.5 נקודות על הימצאו במיקום זה. כלומר, היינו רוצים שהשחקן שלנו ישאף להימצא במשבצות שבשולי הלוח על פני משבצות רגילות במרכז הלוח. המוטיבציה לכך היא שמיקומים אלו מאפשרים לנו לאיים על כלי היריב מבלי להיות חשופים בעצמנו לאיום מצד היריב.

iii. חייל רגיל שמיקומו בשורה הראשונה או האחרונה בלוח (שורה 0 או 7), יקבל 5 נקודות על הימצאו במיקום זה. זאת במטרה 'לעודד' את השחקן שלנו לשמור את כליו בשורת הבסיס ובכך למנוע מהיריב ככל הניתן את היכולת להמליך את כליו או לחילופין 'לעודד' כלים שכבר אינם בשורת הבסיס לנוע לכיוון השורה האחרונה בלוח ולהפוך למלכים. הניקוד עבור הימצאות במיקומים אלו גדול יותר מהניקוד הניתן בסעיף הקודם, במטרה למנוע מהשחקן שלנו לבצע מהלכי סרק שיזיזו חייל מהשורה הראשונה לשולי הלוח (לדוגמא ממיקום (0,1) למיקום (1,0)) מה שיפתח צוהר ליריב ויקל עליו להמליך את כליו.

iv. מלך מעצם היותו מלך יקבל 5 נקודות ללא תלות במיקומו על הלוח. המוטיבציה לכך היא מצד אחד 'לעודד' את השחקן שלנו להמליך כמה שיותר חיילים רגילים, ומצד שני לא להתחשב במיקומו של המלך על הלוח במטרה לעודד את השחקן שלנו שלא לשמור את המלך בקו הבסיס, אלא לצאת להתקפה (ראה בהמשך פרמטר נוסף (d) בו אנו מעודדים התקרבות של המלך לחיילי היריב ואיום עליהם).

v. כאשר היריב נותר עם שני חיילים רגילים או פחות - חייל רגיל שלנו מקבל ניקוד עבור מיקומו בהתאם למרחקו מהשורה האחרונה (ככל שהחייל קרוב יותר לשורה האחרונה המתאימה לו, הניקוד עבור הימצאו במיקום זה גדל). ההיגיון הוא שבמצב הלוח הנוכחי אין עוד משמעות להתבצרות כלי השחקן שלנו במקומות אסטרטגיים על הלוח ולכן נרצה לקדם את החיילים הרגילים בלוח בכדי להפוך אותם למלכים ובכך, להגדיל עוד יותר את יתרוננו במשחק. במילים אחרות, פרמטר זה מביא לשינוי באסטרטגיה של השחקן שלנו, כתוצאה ממצב הלוח הנוכחי, המתבטא בכך שחיילים רגילים שעד עתה התבצרו במיקומים אסטרטגיים על הלוח כעת יתחילו לנוע קדימה בכדי להפוך למלכים.

הלוגיקות הללו ממומשות בפונקציה `evaluate_position`.
כעת, כל אחד מהשחקנים יכול לקבל ניקוד פר כל כלי שברשותו על פי מיקומו בלוח. את הניקוד שניתן למיקומי הכלים של כל שחקן אנו סוכמים, מחשבים את הניקוד הממוצע בעבור כל שחקן ולבסוף מחשבים את הפרש הממוצעים בין השחקנים, כך שערך חיובי מעיד על מצב לוח טוב יותר עבור השחקן שלנו וערך שלילי מעיד על מצב לוח טוב יותר עבור היריב, כאשר הערך עצמו (חיובי או שלילי) משקף את עוצמת היתרון.
נוסחא: `avg_pos_diff = avg_my_pos – avg_opponent_pos`
כאמור, המוטיבציה בפרמטר זה נועדה 'לעודד' את השחקן שלנו להביא את כליו למיקומים מסוימים בשלבים שונים של המשחק ולהתבצר בהם עד שמצב הלוח מאפשר שינוי באסטרטגיה. פרמטר זה מאפשר לנו להשיג יתרון משמעותי על ה-simple_player בכך שהוא מעודד' את השחקן שלנו לייצר מצבי לוח טובים יותר לעומת ה-ה-simple_player שאינו מתחשב במיקומים אסטרטגיים על הלוח ואינו משנה את התנהגותו כתלות במצב הלוח.

d. מרחק של מלכים מכלי היריב
פרמטר זה נועד 'לעודד' את השחקן שלנו להשתמש במלכים בתבונה ולנצל את החוזק שלהם על הלוח, וכן להימנע ממצבים אופייניים המתרחשים בסוף המשחק בהם המלכים של שני הצדדים מבצעים מהלכי סרק עד לסיום המשחק בתיקו.
תחילה, אנחנו מחשבים את המרחק של כל אחד מהמלכים שלנו מכל אחד משחקני היריב, סוכמים את המרחקים ומנרמלים אותם ע"י חלוקה בכמות המלכים. בכך אנחנו למעשה מקבלים את המרחק הממוצע של המלכים שלנו מכלי היריב.

i. כאשר מספר המלכים שלנו גדול או שווה מזה של היריב, אנו למעשה ביתרון וברצוננו להתקרב לשחקני היריב על מנת לאיים עליהם (ובתקווה גם לממש את האיום). על כן, במצב זה אנו מחשבים את ההפרש הבא `1 – my_total_king_dist` על מנת שהשחקן שלנו ישאף להקטין את המרחק הממוצע. כלומר, להתקרב לכליי היריב, לאיים עליהם ובכך לחתור לניצחון.

ii. כאשר מספר המלכים שלנו קטן מזה של היריב, היריב למעשה נמצא ביתרון ואז אנחנו פשוט משתמשים במרחק הממוצע עצמו - `my_total_king_dist`, על מנת שהשחקן שלנו דווקא ישאף להגדיל את המרחק ובכך לסיים את המשחק בתיקו ולהימנע מהפסד.

פרמטר זה מהווה יתרון גדול על פני הלוגיקה של simple_player שאינה מכילה כל אסטרטגיה המנצלת את עוצמתם של המלכים. במספר רב של הרצות נוכחנו לדעת שהלוגיקה מביאה את השחקן שלנו לניצחון במצבים שונים למשל, כאשר לשני השחקנים יש מספר מלכים על הלוח או כאשר לשחקן שלנו יש מלך וליריב עדיין אין, אז השחקן שלנו מנצל את המלך שברשותו בכדי לתקוף את חיילי היריב.

כל אחד מהפרמטרים שהוצגו לעיל מאפשר לנו לייצג מצבים שונים ואפשריים של הלוח בעת המשחק ועל מנת לקבל ערך utility מסכם עבור ה-state הנוכחי, נתנו לכל אחד מהפרמטרים שתוארו משקל המייצג את חשיבותו של הפרמטר בחישוב ערך ה-utility הסופי.

<u>בכדי לקבוע את ערכי המשקולות ביצענו ניסויים רבים בחנו את תוצאותיהם ובסופם הגענו לצורת המשקול הבאה:</u>

- עבור פרמטר a - ההפרש בין כמות הכלים הכוללת של השחקנים - ניתן משקל 100.
- עבור פרמטר b - ההפרש בין כמות המלכים של השחקנים - ניתן משקל 50.
- עבור פרמטר c - הפרש ממוצעי הציונים על סמך מיקומי הכלים על הלוח - ניתן משקל 1.
- עבור פרמטר d - המרחק הממוצע של המלכים מכלי היריב - ניתן משקל 1.

לסיכום, היוריסטיקה שלנו לא רק מסתכלת על כמות הכלים הרגילים והמלכים שעל הלוח (בדומה ל-simple_player), אלא גם שואפת לתפוס מיקומים אסטרטגיים על הלוח ומשנה את האסטרטגיה הזו עם התקדמות המשחק ובכך משיגה יתרון משמעותי על ה-simple_player שכלל לא לוקח בחשבון מיקומים אסטרטגיים. בנוסף, באמצעות חישובי המרחקים של המלכים מכלי היריב, היוריסטיקה שלנו חותרת לניצחון במצב של יתרון במלכים או מתחמקת באלגנטיות מהפסד במצב של יתרון ליריב, ובכך משיגה יתרון נוסף על פני ה-simple_player עם היכולת לסיים משחק בתוצאה המיטבית מבלי לבצע מהלכי סרק שלרוב מובילים לתיקו.

ניהול זמנים:

7. כאמור, על פי הגישה הנאיבית של simple_player לחלוקת הזמן, עבור כל מהלך מ-k המהלכים יוקצה total_time/k זמן. ה- simple_player משתמש בזמן היחסי שהוקצה לו פר מהלך בכדי להפעיל את אלגוריתם ה-Minimax בשיטת iterative-deepening search ולהעמיק את החיפוש ככל שמתאפשר לו במסגרת זמן זה, אך זאת מבלי לבחון את השיפור בתועלת שכל העמקה נוספת תורמת לו.
בכדי לחלק את הזמן בין k המהלכים בתבונה נרצה כן לבדוק האם כדאי לנו להמשיך את החיפוש ולהעמיק יותר או האם עדיף לנו לשמור את הזמן הזה למהלכים הבאים בהם העמקה נוספת תהיה הכרחית לשם קבלת ההחלטה מושכלת בנוגע מהו המהלך הבא הטוב ביותר שעל השחקן לבצע. לשם כך, החל מסף עומק מסוים (עליו נפרט בהמשך) אנחנו נתחיל לבדוק בסיום כל חיפוש עד עומק limit כלשהו (> סף העומק) מה ערך ה-alpha שחזר ומהו המהלך הבא הטוב ביותר על פי חיפוש עד עומק זה. במידה ובמשך מספר איטרציות (החל מאותו סף עומק), בהן אנו ממשיכים להעמיק, לא חל שינוי בערך ה-alpha והמהלך הבא הטוב ביותר נשאר זהה, אנחנו נעצור את החיפוש ונחזיר את הצעד הבא הטוב ביותר שמצאנו עד כה. בצורה זו, אנחנו מונעים העמקה

נוספת שאינה נצרכת במקרה של המהלך הנוכחי למרות שמסגרת הזמן המוקצית כן מאפשרת זאת, ובעצם חוסכים זמן שאותו נוכל לנצל עבור המהלכים הבאים בהם יתכן ונצטרך להעמיק את החיפוש מעבר למסגרת הזמן המוקצית. אותו סף עומק שהוזכר למעשה מייצג עומק שעד אליו אנחנו מבצעים את החיפוש בצורה סדירה כלומר, ממש כמו ב-simple_player, אך ההבדל הוא שהחל מאותו סף עומק והלאה אנחנו מתחילים להיות ערניים להאם כל העמקה נוספת אכן תורמת לנו להחלטה או לא. ואז במידה ואנחנו רואים שבמשך מספר העמקות נוספות (קבענו מספר זה להיות 3) אין כל שינוי, אזי מבחינתנו אין טעם להמשיך את החיפוש. כלומר, נוכל להחזיר את ההחלטה לגבי מהו המהלך הבא הטוב ביותר רק לאחר שחיפשנו עד סף העומק + 3 איטרציות הבדיקה. העמקה נוספת במסגרת הזמן המוקצה נבצע כל עוד ערך האלפא או זהות המהלך הבא הטוב ביותר ממשיכים להשתנות.

<u>בכדי להחליט מה יהיה אותו סף עומק (אותו הזכרנו קודם) לקחנו בחשבון מספר קריטריונים שעל עומק זה לקיים:</u>

- עד עומק זה + 3 איטרציות הבדיקה, החיפוש צריך להתבצע בצורה מהירה יחסית בכדי שלא לבזבז זמן מיותר.
- חיפוש עד עומק זה + 3 האיטרציות הנוספות יכול להביא להחלטה מושכלת דיו עבור מהו המהלך הבא הטוב ביותר שעל השחקן לבצע.

מבדיקה שערכנו נראה שחיפוש עד שכבה 4 (כולל) הוא מאוד מהיר (מאיות שנייה) והמשך החיפוש לעוד 3 איטרציות אינו מבזבז יותר מדי זמן. בנוסף, מצאנו שעומק 7 כולל (=4+3) מאפשר קבלת החלטה מושכלת דיו לגבי מהו המהלך הבא הטוב ביותר שעל השחקן לבצע.

כעת, לאחר שמצאנו דרך לחסוך בזמן היינו רוצים גם לאפשר עבור מהלכים מסוימים הזקוקים לכך את האפשרות להעמיק מעבר לעומק המקסימאלי המתאפשר במסגרת הזמן המוקצה לאותו מהלך. לשם כך אנחנו משתמשים בפונקצית `selective_deepening_criterion` המאפשרת לנו להעמיק את החיפוש עבור מהלך מסוים בהתאם לקריטריונים מסוימים. בחרנו להעמיק את החיפוש במידה ואנחנו רואים שהמהלך הבא הוא פעולת קפיצה (capture), שכן פעולה זו יכולה להיות קריטית לבחירתנו את המהלך הבא והיינו רוצים להעמיק בכדי לקבל תובנות לגבי מצב הלוח המתקבל לאחר פעולת קפיצה זו.

<u>אם נסכם זאת באופן פורמאלי, השיטה האינטליגנטית שלנו לחלוקת הזמן של השחקן בין כל k המהלכים היא:</u>

i. בדומה ל-simple_player, הקצה לכל תור מסגרת זמן לפי הנוסחה הבאה – `(Remaining Time / Remaining Turns) – 0.05`.

ii. בצע חיפוש בצורה סדירה - שכבה אחר שכבה החל מעומק limit = 1 ועד עומק limit = 4 כולל.

iii. החל מעומק limit = 5, בדוק עבור כל העמקה נוספת שכזו האם יש שיפור בערך ה-alpha והאם המהלך הבא הטוב ביותר השתנה בעקבות החיפוש עד העומק limit הנוכחי.

iv. במידה ובמשך 3 איטרציות רצופות לא היה שינוי בערך ה-alpah והמהלך הבא הטוב ביותר לא השתנה בעקבות העמקות אלו, עצור את החיפוש והחזר את המהלך הבא הטוב ביותר שנמצא עד כה.

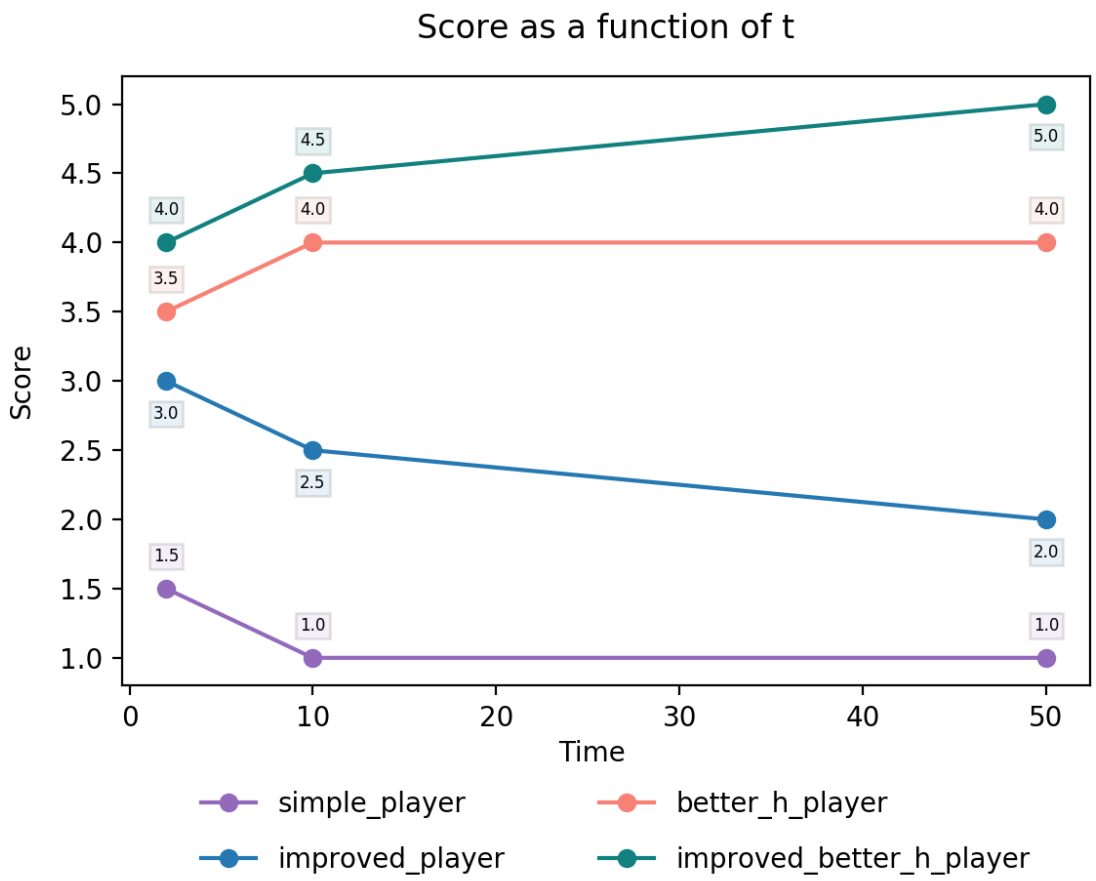
v. אם ניצלת את כל מסגרת הזמן המוקצה למהלך הנוכחי, אך המהלך הבא הוא פעולת קפיצה, אזי המשך להעמיק את החיפוש.

vi. במידה והמהלך הנוכחי הוא המהלך ה-k (מתוך k) כלומר, המהלך האחרון בסבב k המהלכים, אזי נצל עד תום את כל הזמן שנותר מ-k-1 המהלכים הקודמים יחד עם הזמן שהוקצה למהלך זה כחלק מחלוקת הזמן המקורית.

9. מצורף בתיקיית קבצי הקוד.

10.

i.



ii.

	t = 2	t = 10	t = 50
simple_player	1.5	1	1
improved_player	3	2.5	2
better_h_player	3.5	4	4
improved_better_h_player	4	4.5	5

iii. השחקנים עם היוריסטיקה הפשוטה: simple_player, improved_player
השחקנים עם היוריסטיקה שהוצעה בשאלה 6: better_h_player, improved_better_h_player

ממבט על על הגרף ממש אפשר לראות שהביצועים של השחקנים עם היוריסטיקה הפשוטה הם מעין תמונת מראה הפוכה של הביצועים של השחקנים עם היוריסטיקה המשופרת. במילים אחרות, העקומה המתארת את ביצועי ה- simple_player היא תמונת מראה הפוכה של העקומה המתארת את ביצועי better_h_player והעקומה המתארת את ביצועי improved_player היא תמונת מראה הפוכה של העקומה המתארת את ביצועי improved_better_h_player. כלומר, מגמות השיפור והדעיכה הן ממש הפוכות - כאשר יש מגמת שיפור בביצועים של השחקנים עם היוריסטיקה המשופרת במגבלת זמן מסוימת אז בהתאמה הביצועים של השחקנים עם היוריסטיקה הפשוטה נמצאים במגמת דעיכה. באופן כללי, ניתן לראות מהגרף שביצועי השחקנים עם היוריסטיקה המשופרת רק עולים ככל שמגדילים את מגבלת הזמן, למעט ביצועי ה- better_h_player עבור t = 50 אשר זהים לביצועיו במגבלת הזמן של t = 10, אך עדיין אין כל מגמת דעיכה עבור שחקנים אלו - רק שיפור או שמירה על הביצועים בין מגבלות הזמן השונות. לעומת זאת, השחקנים עם היוריסטיקה הפשוטה נמצאים באופן תמידי במגמת דעיכה, בדומה לסייג מעלה עבור ה-better_h_player גם כאן יש סייג והוא שביצועי ה- simple_player עבור t = 50 זהים לביצועיו במגבלת הזמן של t = 10. כלומר, ביצועיו נשמרו בין מגבלות זמן ולא הורעו, אך גם לא השתפרו. אך באמור, המגמה הכללית של השחקנים עם היוריסטיקה הפשוטה היא מגמת דעיכה.

<u>כעת אפרט באופן יותר נרחב על הביצועים של השחקנים בכל מגבלת זמן:</u>

- <u>עבור t = 2</u>
  במגבלת זמן זו, קל לראות שלשחקנים עם היוריסטיקה הפשוטה יש מראש נקודת פתיחה גרועה ביחס לשחקנים עם היוריסטיקה המשופרת - ניקוד כולל של 1.5 ו- 3 נקודות ל- simple_player וה- improved_player בהתאמה, לעומת ניקוד כולל של 3.5 ו- 4 נקודות ל- better_h_player וה- improved_better_h_player בהתאמה. רק מהסתכלות על הניקוד הכולל של simple_player ושל better_h_player, במגבלת זמן זו, שכל השוני ביניהם מסתכם ביוריסטיקה שונה, קל להבחין שמדובר בשיפור ביצועים משמעותי ביותר בעקבות יישום היוריסטיקה המשופרת במקום היוריסטיקה הפשוטה והנאיבית - הפרש של 2 נקודות יותר בניקוד הכולל לטובת השחקן עם היוריסטיקה המשופרת - better_h_player.

- <u>עבור t = 10</u>
  במגבלת זמן זו, אנחנו רואים שביצועי השחקנים עם היוריסטיקה המשופרת במגמת שיפור בהשוואה לביצועיהם במגבלת הזמן של הקודמת - שיפור של חצי נקודה בניקוד הכולל לכל אחד מהשחקנים. כעת, improved_better_h_player עם ניקוד כולל של 4.5 נקודות ו- better_h_player עם ניקוד כולל של 4 נקודות. לעומת זאת, ביצועי השחקנים עם היוריסטיקה הפשוטה דווקא נמצאים במגמת דעיכה לעומת ביצועיהם במגבלת הזמן הקודמת - חצי נקודה פחות בניקוד הכולל לכל אחד מהשחקנים. כעת, improved_player עם ניקוד כולל של 2.5 נקודות ו- simple_player עם ניקוד כולל של נקודה אחת.

- <u>עבור t = 50</u>
  מבחינת השחקנים עם היוריסטיקה המשופרת אנחנו רואים ש- improved_better_h_player ממשיך במגמת שיפור עם תוספת של חצי נקודה בניקוד הכולל בהשוואה למגבלת הזמן הקודמת. כלומר, כעת ל- improved_better_h_player ניקוד כולל של 5 נקודות. לעומתו ה- better_h_player שומר על ביצועיו ממגבלת הזמן הקודמת, כאשר נשאר עם ניקוד כולל של 4 נקודות. עם זאת, מבחינת השחקנים עם היוריסטיקה הפשוטה אנחנו רואים שביצועי ה- improved_player ממשיכים במגמת דעיכה עם חצי נקודה פחות בניקוד הכולל מהניקוד שהשיג במגבלת הזמן הקודמת. כלומר, כעת ל- improved_player ניקוד כולל של 2 נקודות. לעומתו, ה- simple_player (בדומה ל-better_h_player) שומר על ביצועיו ממגבלת הזמן הקודמת, כאשר נשאר עם ניקוד כולל של נקודה אחת.

באופן, כללי, ניתן לראות שהשחקנים כל אחד עם היוריסטיקה המושמת בו ממשיכים באותה מגמה שהייתה להם במגבלת הזמן הקודמת או שומרים על ביצועיהם ממגבלת הזמן הקודמת.

לסיכום:
לאורך כל הדרך (בכל מגבלות הזמן), היוריסטיקה שלנו מתגברת על היוריסטיקה הפשוטה ואף בפער לא מבוטל וזאת כי היוריסטיקה שלנו לוקחת בחשבון מספר פרמטרים (עליהם פירטנו בהרחבה בתשובתנו לשאלה 6) בחישוב ה-utility של state מסוים של הלוח - ההפרש בין מספר הכלים הכולל של השחקנים, ההפרש בין מספר המלכים של השחקנים, מיקומים אסטרטגיים על הלוח ע"י מתן ציונים למיקומים של כל כלי על הלוח עבור כל שחקן והפרמטר האחרון שהוא חישוב המרחק הממוצע של המלכים מכלי היריב 'המעודד' תקיפה של המלכים את חיילי היריב. הפרמטרים הללו מאפשרים התמודדות עם מצבי לוח שונים ושינוי התנהגות או אסטרטגיה כתלות במצב הלוח ושקלול כל הפרמטרים הללו מאפשר לנו לכמת את כל המידע הזה לציון utility אחד שבסופו של דבר עוזר לנו לבחור בצורה מושכלת את המהלך הבא שלנו במשחק באופן שמסתכם ברוב המקרים לניצחון במשחק, מה שמגדיל את תוחלת הניצחונות של השחקנים המשתמשים ביוריסטיקה המשופרת הזו לעומת תוחלת הניצחונות של שחקנים אשר ישתמשו ביוריסטיקה הפשוטה והנאיבית - בה ה-utility מחושב באמצעות שקלול כמות החיילים והמלכים שיש לכל שחקן על הלוח ותו לא.
נשים לב, שבכל שאנחנו מגדילים את מגבלת הזמן היוריסטיקה המשופרת שלנו מצליחה להביא את השחקנים המשתמשים בה לביצועים יותר טובים או לפחות לשמירה על ביצועיהם הגבוהים ממגבלות הזמן הקודמות, זה רק מוסיף לעובדה שהיוריסטיקה המשופרת שהצענו היא אכן מוצלחת, לעומת היוריסטיקה הפשוטה שהניקוד הכולל של השחקנים המשתמשים בה יכול למעשה רק להרע.

iv. השחקנים בלי השיפור בניהול הזמנים: better_h_player & simple_player
השחקנים עם השיפור בניהול הזמנים: improved_player & improved_better_h_player

פירוט על הביצועים של השחקנים בכל מגבלת זמן:

- עבור t = 2
במגבלת זמן זו, קל לנו לראות את ההבדלים בביצועים עקב מנגנון ניהול הזמנים השונה כאשר מסתכלים על הניקוד הכולל של simple_player לעומת הניקוד הכולל של improved_player, שכן כל השוני ביניהם מסתכם במנגנון ניהול זמנים שונה. אם כן, ניתן לראות שהשימוש של improved_player במנגנון ניהול הזמנים המשופר שהגינו בשאלה 7 מביא לביצועים פי 2 יותר טובים מאלו של simple_player אשר משתמש במנגנון ניהול הזמנים הלא משופר - ניקוד כולל של 3 נקודות ל- improved_player לעומת ניקוד כולל של 1.5 נקודות ל- simple_player.
אם נסתכל על הביצועים של better_h_player לו מנגנון ניהול זמנים לא משופר ועל הביצועים של improved_better_h_player אשר משתמש במנגנון ניהול זמנים משופר, במסגרת מגבלת זמן זו, אפשר להסיק שהגורם העיקרי לכך שלשניהם ביצועים מאוד טובים, יחסית לשחקנים האחרים, הוא השימוש ביוריסטיקה המשופרת משאלה 6, אך עדיין ניתן לשים לב שהשימוש במנגנון הזמנים המשופר תורם ל- improved_better_h_player תוספת של חצי נקודה בניקוד הכולל. אמנם יש שיראו תרומה זו כזניחה, אך תרומה זו מספיקה בכדי להראות שמנגנון ניהול הזמנים המשופר אכן תורם לשיפור הביצועים.

- עבור t = 10
במגבלת זמן זו, אם שוב נשווה בין הביצועים של simple_player לו מנגנון ניהול זמנים לא משופר לעומת הביצועים של improved_player אשר משתמש במנגנון ניהול זמנים משופר, נשים לב שאמנם שניהם במגמת דעיכה, אך עדיין הביצועים של improved_player טובים בלפחות פי 2 מביצועי simple_player - ניקוד כולל של 2.5 ל- improved_player לעומת נקודה אחת ל- simple_player.
כעת, אם נשווה את הביצועים של better_h_player המשתמש במנגנון ניהול זמנים שאינו משופר עם הביצועים של improved_better_h_player המשתמש במנגנון ניהול הזמנים המשופר, נראה ששניהם נמצאים במגמת שיפור ביחס לביצועיהם במגבלת הזמן הקודמת - שיפור של חצי נקודה לכל אחד מהם בניקוד הכולל. כלומר, כעת הניקוד הכולל של improved_better_h_player הוא 4.5 נקודות והניקוד

הכולל של better_h_player הוא 4 נקודות. כמובן שבהשוואה ל- improved_player וה- simple_player ברור לנו שהשיפור בביצועים של better_h_player ושל improved_better_h_player נובע מהיוריסטיקה המשופרת, אך בדומה למגבלת הזמן הקודמת, עדיין יש לנו את השיפור הקטן המתאפיין בתוספת חצי נקודה לניקוד הכולל של improved_better_h_player לעומת הניקוד הכולל של better_h_player, כתוצאה מהשימוש של improved_better_h_player במנגנון ניהול הזמנים המשופר שלנו.

- <u>עבור t = 50</u>

במגבלת זמן זו, אם נשווה בין הביצועים של simple_player לו מנגנון ניהול זמנים לא משופר לעומת הביצועים של improved_player אשר משתמש במנגנון ניהול זמנים משופר, אנחנו יכולים לראות ש- improved_player ממשיך במגמת דעיכה עם חצי נקודה פחות בניקוד הכולל בהשוואה לביצועיו במגבלת הזמן הקודמת. אמנם לעומתו, simple_player שומר על ביצועיו ממגבלת הזמן הקודמת. עם זאת, עדיין הביצועים של improved_player טובים פי 2 מהביצועים של simple_player - ניקוד כולל של 2 ל- improved_player לעומת נקודה אחת ל- simple_player.
כעת, אם נשווה את הביצועים של better_h_player המשתמש במנגנון ניהול זמנים שאינו משופר עם הביצועים של improved_better_h_player המשתמש במנגנון ניהול הזמנים המשופר, במסגרת מגבלת זמן זו, נראה ש- improved_better_h_player ממשיך במגמת שיפור עם תוספת של חצי נקודה לניקוד הכולל ביחס לביצועיו במגבלת הזמן הקודמת - ניקוד כולל של 5 לעומת ניקוד כולל של 4.5 במגבלת הזמן הקודמת. לעומתו, better_h_player שומר על ביצועיו ממגבלת הזמן הקודמת - 4 נקודות. כלומר, כאן אנו מקבלים שהתרומה של מנגנון ניהול הזמנים המשופר לניקוד הכולל של improved_better_h_player היא עוד נקודה שלמה.

<u>לסיכום:</u>
גם כאן, כמו בסעיף הקודם, בכל מגבלות הזמן לעיל, מנגנון ניהול הזמנים המשופר שלנו מביא לשיפור בביצועי השחקנים. כמובן שהצהרה זו נובעת כאשר אנו משווים לחוד בין ביצועי ה- simple_player לביצועי ה- improved_player ובין ביצועי ה- better_h_player לביצועי ה- improved_better_h_player, שכן בצורה כזו אנו יכולים לבודד את התועלת ששימוש במנגנון ניהול הזמנים המשופר נותן לנו. הסיבה לשיפור זה היא שמנגנון ניהול הזמנים המשופר שלנו עובד בצורה ייחודית המאפשרת לו לחלק בצורה יעילה ודינאמית את הזמן בין k המהלכים בכל סבב של k מהלכים, כל זאת באמצעות הבדיקה שאנחנו מבצעים החל מסף עומק מסוים אשר מטרתה לחסוך בזמן המתבזבז על העמקה מיותרת עבור מהלכים מסוימים וניצול זמן זה להעמקה מוגברת במהלכים עבורם העמקה שכזו נצרכת בכדי להגיע להחלטה מושכלת. בנוסף, אנו מאפשרים לחרוג ולהעמיק מעבר לזמן המוקצה למהלך ספציפי במידה ואנחנו מזהים שהמהלך הבא הוא פעולת קפיצה (capture), שכן תוצאות מהלך זה מאפשרות לנו את שקלול מצב הלוח לאחר פעולה זו והסקת מסקנות חשובות שיכולות להשפיע על המהלכים הבאים שלנו במשחק. לעומת זאת, מנגנון ניהול הזמנים הלא משופר לא מביל כל לוגיקה שיכולה להעיד על מצבי לוח עבורם העמקה נוספת הייתה יכולה מאוד לעזור להמשך המשחק. בנוסף, המנגנון הפשוט אף מאפשר העמקה שלא לצורך ובזבוז זמן משווע שברוב המקרים יביא להפסד במשחק.

<u>הערת מימוש:</u>

במסגרת מימוש היוריסטיקה נעזרנו ברעיונות שמצאנו באינטרנט ממקורות שונים:

1. נעזרנו במימוש הבא לצורך פרמטר המיקומים האסטרטגיים על הלוח, תוך ביצוע <u>שינויים משמעותיים</u> בלוגיקה אשר מומשה שם. חלקים נרחבים מהלוגיקה, כגון: הפרדה בין שולי הלוח לשורות הקדמיות והאחוריות ושינוי הלוגיקה כאשר המשחק נמצא בשלב מתקדם (ליריב אין יותר מחייל רגיל אחד) לא מופיעים במימוש המקורי ופותחו על ידינו. https://github.com/billjeffries/jsCheckersAI/blob/master/scripts/checkers-engine.js

2. שאבנו רעיונות מימושיים שונים (רעיונות עקרוניים שאינם כתובים בקוד), כגון שימוש במרחקי המלכים מכלי היריב, מהקישור הבא: https://www.cs.huji.ac.il/~ai/projects/old/English-Draughts.pdf