**תרגיל בית 2 - Checkers**

**חלק א':**

1. כל מצב במרחב המצבים הינו סידור הלוח באותו רגע נתון. כלומר, כל מצב הינו הלוח עצמו בו ממוקמים חיילי היריבים באותו תור, וכן מי השחקן שתורו לשחק עכשיו וכמה תורות עברו מאז הקפיצה האחרונה (על מנת להכריע האם יש תיקו במשחק).

2. ייצוג אופרטור, על ידי המחלקה GameMove, נעשה באמצעות 3 פרמטרים – החייל עצמו (איזה צבע והאם חייל פשוט או מלך), מהו המקום ממנו הוא צועד, ומהו המקום אליו יזוז. בנוסף ישנו שדה המאפשר לנו להבחין בין תנועה רגילה לפעולת קפיצה. כאשר מתבצעת תנועה רגילה השדה jumped\_locs יהיה ריק או None, ואילו בפעולת קפיצה שדה זה יכיל את כל הקפיצות שנעשו.

**חלק ב':**

3. הגישה ה"נאיבית" בה נוקט השחקן לשם חלוקת הזמן בין כל K המהלכים היא חלוקה אחידה של הזמן לכל הצעדים פחות buffer קבוע. כלומר, כלומר כל צעד יקבל כמות זמן זהה.

4. חסרון אפשרי של השיטה הזו היא שבגלל שהוא מקציב זמן זהה עבור כל תור השחקן לא יצליח לפתח עד עומק מאוד עמוק בעץ בכל איטרציה , נקרא לעומק הזה d ולכן אם יש שחקן חכם שיגרם לו ל"הרגשת" ניצחון בעומק d (שזה העומק שsimple\_player נניח יצליח לפתח) ומכין לו הפתעה בd+1 (כלומר שהשחקן החכם פתאום ייתן מהלך שיאכל הרבה שחקנים של simple\_player) הוא יוכל לנצח אותו.

5. הפונקציה היוריסטית שמשתמש בה השחקן היא:

הוא מחשב כמה שחקנים "פשוטים" (=לא מלכים) יש לו וכמה יש למתמודד השני. בנוסף, מחשב כמה מלכים יש לו וכמה למתמודד. אם לאויב אין שחקנים (כלומר ניצחון לו) מחזיר ∞ (או במקרה שלנו 6000). לעומת זאת, אם אין שחקנים לו (כלומר ניצחון לאויב) מוחזר -∞ (או במקרה שלנו -6000).

בכל מקרה אחר, ממשקלים את השחקנים שעל הלוח, משקל שחקן פשוט הוא 1 ושחקן "מלך" הוא 1.5 ומה שמוחזר הוא our\_weight – opponent\_weight.

או בצורה פורמלית:

נגדיר משקל שחקן כ:

המוטיבציה בפונקציה היורסטית הזו:

בכל תור על השחקן להעריך כמה מצבו הוא טוב ואיזה צעד (פעולה) כדאי לו לבצע.  
במקרה שרק לו יישארו שחקנים על הלוח (=נצחון) יוחזר ∞ כדי להמשיך בכיוון זה.  
לעומת זאת, במצב בו רק לאויב יישארו לאויב שחקנים (=הפסד) יוחזר ∞- כדי לא לפתח בכיוון זה.

בכל מצב אחר, לשני השחקנים יש כלים על הלוח \*אבל\* לא כל השחקנים בעלי כוח זהה (למלך יש יותר פונקציונאליות מלשחקן פשוט) ולכן הערכת מצב המשחק צריכה להתחשב גם בכוח השחקן.  
לכן, ההערכה עד כמה המצב טוב לשחקן הוא "משקל" השחקנים שלו פחות "משקל" שחקני היריב כך ש"משקל השחקנים" של שחקן מוגדר כחוזק של מלך (1.5) כפול מספר המלכים ועוד חוזק של חייל פשוט (1) כפול מספר החיילים הפשוטים שלו.

על ידי הפונקציה הזו, שחקן יכול להעריך כמה מצבו טוב והאם הוא מוביל בstate.

**חלק ג:**

6.

1. ההיוריסטיקה שנגדיר עבור הערכת מצבי המשחק היא במספר נקודות:
   1. מתייחסת בדומה לSimple Player ל2 סוגי החיילים במשחק.
   2. מנרמלת לתוכה את מיקום החיילים הפשוטים על הלוח.
   3. צידי הלוח: חייל פשוט הנמצא בצד הלוח הוא "מוגן" מפני אכילה ולכן מקבל ניקוד נוסף. לעומת זאת, מלך שנמצא בפינת הלוח נתון יותר למניפולציות היריב (כלומר קל יותר לאכול אותו ולהכין מהלך שינצח אותו – כי מספר האפשרויות שלו לזוז קטנות יותר).
   4. אסטרטגיית אכילת חיילים פשוטים ע"י מלכים.
   5. כאשר את לוח המשחק מאיישים רק מלכים יש 2 מצבים: מצב בו אנחנו מובילים ולכן נרצה להתקרב אל היריב , או במידה ויש לו יותר מלכים נשאף לתיקו ונתרחק מהמלכים של היריב.

לצורך הנוחות נגדיר מספר פונקציות:

* *פונקציית הערכה עבור שחקני הלוח ומשקלם (ביחס לכוחם):*

*(\*) הפונקציה rowsMoved היא מספר השורות שכל אחד מהחיילים עבר (במידה והוא אדום אז זה i במידה ושחור 7-i).*

* *פונקציית הערכה עבור מיקום החיילים הפשוטים (המוטיבציה שלהם להגיע לצד השני ובו להפוך למלכים):*
* *פונקציית הערכה עבור מלכים שנמצאים בפינות (וכך קל יותר להכין להם מלכודת):*
* *פונקציית הערכה עבור חיילים פשוטים שנמצאים בפינות (ולא ניתנים לאכילה):*
* *אסטרטגיה לאכילה של חיילים פשוטים ע"י מלכים:*
* אסטרטגיה למצב בו יש רק מלכים:

(\*) הפונקציה dist מחשבת את המרחק בין שני הכלים, במידה ולשחקן יש יותר כלים אז שואפים להתקרב אליו במדיה ויש לנו פחות כלים אז נרצה להתרחק.

חישוב מרחק בין 2 רכיבים בלוח:

כאשר יש לנו יותר מלכים שנאף להתקרב ליריב , במידה ולא נשאף להתרחק:

1. המוטיבציה להגדרה זו היא הבנה נוספת של המצב האמיתי במגרש, מספר החיילים בלוח ומשקלם (simpleEvaluation) הוא אכן פקטור מרכזי בהבנה שלנו על המצב בלוח עם זאת, הוא לא הפרמטר היחיד.

בפונקציה ההיוריסטית שלנו הגדרנו עבור המצב הפשוט גם את מיקום החיילים הפשוטים (positionEvaluation) בלוח הרי חיילים שקרובים להפוך למלכים עדיפים על חיילים שנמצאים מאחורה ולכן כדאי לשקלל זאת בפונקציה.

בנוסף, חיילים פשוטים שנמצאים בפינות הלוח מוגנים מפני אכילה ולכן מגיע להם תוספת קטנה בניקוד על "ישיבה" שם (pawnInCorner).

בנוסף, כל עוד לא נותרו רק מלכים כדאי שהמלכים יתעסקו בלאכול את החיילים הפשוטים היריבים (KingCatchPawn).

עם זאת, ברגע שהמשחק נהיה רק של מלכים הutility צריך מעט להשתנות – עדיין ניקח בחשבון את simpleEvaluation (ההיוריסטיקה הפשוטה שמכילה את מספר החיילים ומשקלם) בנוסף אליה גם במידה ואנחנו מובילים (יש לנו יותר מלכים) או מספר מלכים שווה נשאף שהמלכים יתקדמו לכיוון מלכי האויב כדי להכריע אותם ואם פחות נדאף לתיקו ונברח מהמלכים של האויב (sumOfDist).

כל הפרמטרים האלו ישפרו את ההיוריסטיקה כי בזכותה לא רק מתחשבים במספר החיילים בלוח אלא בפרמטרים נוספים ובאמצעותה ניתן יהיה לקצץ ענפים בעץ בצורה טובה יותר (וכך גם לסרוק יותר לעומק) ובכוונון הפרמטרים הנכון להשליך את המצב במשחק בצורה טובה יותר.

7. השיטה הנאיבית הנתונה במשחק, כאשר יש לכל k מהלכים t זמן, מחלקת לכל מהלך k/t (פחות 0.05 שניות) זמן. ניתן להגדיר חלוקה אחרת של הזמן על מנת לשפר את השחקן הפשוט, באופן הבא:

כאשר k קטן או שווה ל2, אזי נשמור על השיטה הנאיבית, כי ב-2 תהליכים אין כ"כ הרבה הבדל אם הזמן שלהם זהה.

כאשר k גדול מ-2, החלוקה תהיה באופן הבא – המהלך הראשון יקבל t/2 מהזמן, המהלך השני יקבל t/3 מהזמן, השלישי יקבל t/4 מהזמן וכך הלאה.

המוטיבציה בחלוקה הזו נובעת מהשימוש ב-minimax. לכל מצב במשחק יש תת עץ אותו צריך לחשב על מנת לבחור מה יהיה הצעד הבא הטוב ביותר. ההנחה היא שככל שאנחנו נמצאים בשלב מוקדם יותר במשחק, ישנו מרחב אפשרויות יותר גדול ולכן תת העץ לצעד הבא גדול יותר, וממילא דורש יותר זמן חישוב. לכן, חלוקת הזמן תהיה באופן הזה, בו למהלך הראשון יהיה חצי מהזמן, לשני יהיה שליש מהזמן, לשלישי רבע וכך הלאה.

באופן כזה, אנחנו גם לוקחים מקדם ביטחון ( בדומה ל-0.05 שניות שהורידו בשיטה הנאיבית), כי סכום הסדרה שאנחנו יוצרים כאן לא מגיעה ל-t זמן. (שהרי חצי ועוד שלישי ועוד רבע וכו' שואף ל-1 אבל לא שווה לו).