**מבוא לבינה מלאכותית – תרגיל בית 2 – פאקמן:**

1. **הכרות עם הקוד והמשחק:**
   1. שחקן בסיסי ReflexPlayer בוחר בכל צומת החלטה את הפעולה הבאה לביצוע ע"י בחינת המצבים הבאים אליהם כל פעולה אפשרית מובילה, ובחירת הפעולה המובילה למצב בעל הניקוד הגבוה ביותר. פונ' היוריסטיקה היא, אם כן, הניקוד המקס' של המצב המבין סט המצבים הבאים
2. **בניית סוכן משופר:**
   1. היוריסטיקה שלנו תהא מורכבת מקומבינציה לינארית של המאפיינים הבאים (הפרמטרים ניתנים לכוונון):
      * ניצחון/הפסד
      * מרחק מינימלי לאוכל
      * ניקוד
      * כמות האוכל במגרש
      * כמות הקפסולות במגרש
      * האם קפסולה
      * כמות הרוחות הרעות במגרש
   2. החישוב שלנו עבור היוריסטיקה מובא להלן: 
      * יש לשים לב למקדמים השליליים עבור חלק מהפרמטרים. הם קיימים על מנת לאפשר לוגיקה הפוכה; המצב נבחר כמקסימלי, לכן נרצה לתת ערך גבוה ככל האפשר עבור:
        1. כמות אוכל קטנה יותר – נבחר במצב בו אוכלים על פני מצב בו לא אוכלים.
        2. מרחק מינימלי לאוכל – נבחר להתקדם לכיוון האוכל הקרוב ביותר על פני כיוון האוכל הרחוק.
        3. במקרה בו המשבצת הסמוכה אלינו מכילה קפסולה, נבחר להתקדם אליה על פני משבצת בה אין קפסולה.
        4. נבחר להתקדם למשבצת בה מס' הרוחות הפעילות נמוך יותר, כלומר למשבצת בה ביצענו אכילה של רוח רעה, על פני משבצת בה לא ביצענו זאת.
      * ניתן היה לממש יוריסטיקה עוד יותר טובה באמצעות monkey hacking קטן לסוכן עצמו, אבל בחרנו שלא לעשות זאת. העובדה הזו נובעת מהאופן בו מומש הסימולטור, בייחוד במשבצות בהן גם אוכלים קפסולה, ומיד באותו צעד הרוחות משתנות לרוחות טובות.

הדבר גרם לנו להתנגשות של 2 יוריסטיקות מתחרות – השאיפה לאכול קפסולה, והשאיפה לנוע לכיוון רוחות טובות.

ניתן היה למנוע זאת באמצעות שמירת אינדיקטור על אכילת הקפסולה, כך שיופעל סעיף היוריסטיקה של תנועה לכיוון רוחות טובות, רק לאחר אכילת הקפסולה.

1. **בניית סוכן Min-Max**
   1. על ידי שימוש בעץ אסטרטגיה, אנו למעשה מניחים כי:
      * קבלת החלטות במשחק מתקבלת בתורות. פעם תור הסוכן ובפעם תור כל אחת מהרוחות. כלומר, אנו מניחים שרוח לא תקבל החלטה פעמיים בלי שהסוכן קיבל החלטה פעם אחת באמצע. הנחה זו לא בהכרח נכונה שכן קבלת ההחלטות מתקבלת אצל הסוכנים השונים במקביל.
      * השחקן היריב גם משחק בצורה אופטימלית.
2. דרך נוספת לחישוב ערך minimax היא להתייחס לכל הרוחות יחד כסוכן אחד. בדרך זו נשארת שכבת min אחת בעץ האסטרטגיה.
   * + חסרון שיטה זו בא לידי ביטוי במקדם הסיעוף. בעץ בעלת שכבתmin אחת לכל רוח, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולת הרוח הספציפית. בעץ בעל שכבת min אחת בלבד, מרחב המצבים השכנים יכלול את המצבים הבאים לאחר פעולה של כל אחת מהרוחות ולכן זהו מרחב מצבים גדול בצורה ניכרת, כלומר מקדם הסיעוף B גדול במצב זה משמעותית.

לפי משפט מהתרגול, סיבוכיות הזיכרון היא . אפילו שעומק העץ הצטמצם, כיוון שמקדם הסיעוף גדל כל כך, חיסרון השיטה היא בכמות הזיכרון הגדולה הנדרשת לי*ישומה לעומת השיטה של עץ בעל כמה שכבות min.*

* + - *יתרון שיטה זו בא לידי ביטוי בעומק העץ. לפי משפט מהתרגול סיבוכיות הזמן אקספוננציאלית בעומק - . אפילו שמקדם הסיעוף גדל, עצם העובדה שהצלחנו להקטין את עומק העץ תביא לשיפור בזמן ריצת התוכנית לעומת שיטה המשתמשת בעץ אסטרטגיה בעל כמה שכבות min.*

1. **בניית סוכן Alpha-Beta**
   1. מבנה העץ החדש ישפיע על אלגוריתם alpha-beta באותו אופן שבו השפיע על אלגוריתם minimax. באלגוריתם זה יש שימוש בשני חסמים, (חסם תחתון מקסימלי) ו- (חסם עליון מינימלי) שיאפשרו צמצום של הצמתים שנפתח בעץ. נפתח רק צמתים שיקיימו בכל זמן נתון *.*

*גיזום : פיתוח צומת min. אם* החסם של גבוה מהערך המתקבל באחד השכנים נבצע גזימה *של הענף כיוון שמובטח לנו שערכו של צומת ה-max העליונה יהיה בתחום ולכן אין צורך לפתח ענף המביא לערכים שלא בטווח. אמנם בעץ זה מעל צומת min אין בהכרח צומת max כמו בעץ אסטרטגיה של שני משתתפים, אך לפי הגדרת העומק אנו יודעים כי בסוף כל שכבות ה-min של הרוחות, תמצא שכבת max של פקמן. לכן גם בעץ זה אין טעם לפעפע מעלה מסלולים שבמילא לא יתחשבו בהם בשכבת ה-max הבאה.*

*גיזום : פיתוח צומת max. אם החסם של נמוך יותר מהערך המתקבל באחד השכנים, נבצע גזימה של הענף. אין צורך לפתח ענף שבודאות לא יבחר בשכבת ה-min הבאה.*

*מאותם שיקולים כמו בגיזום , למרות שמבנה העץ שונה, מובטח לנו שאחרי כל שכבת max, תהיה שכבת min ולכן נכונות הגיזום בפרט והאלגוריתם בכלל, נשמרת.*

1. *סוכן ה-alpha-beta יתנהג שונה מסוכן minimax בשני האופנים:*
   * + *זמן הריצה יתקצר – זמן חישוב הצעד הבא יתקצר שכן עבור אלגוריתם זה אין טעם לחשב את ערך minimax עבור כל המסלולים, אלא מסלולים שיודעים מראש שלא יבחרו לא נלקחים בחשבון. מספר החישובים קטן לעומת אלגוריתם minimax וכתוצאה מכך גם זמן הריצה.*
     + *ערך ה-minimax הסופי יישאר זהה לזה של סוכן ה-minimax, אך תתכן בחירת מהלכים שונה וזאת בעקבות מצבים של שוויון בין ערכי minimax. עבור אלגוריתם זה אם יש שוויון וערך minimax של צומת שווה ל- או ל- , הענף נגזם והצומת בוודאות לא תיבחר לפעפע הלאה, זאת בניגוד לאלגוריתם minimax שיכל היה לבחור אחרת.*
2. **בניית סוכן Expectimax לרוח רנדומלית**

כעת יש בידנו מידע נוסף, תנועת הרוחות נבחרת על ידי התפלגות אחידה על כל האפשרויות. Minimax ו-alpha-beta שניהם מניחים כי היריב יעשה את הצעד הכי טוב בשבילו וכאן אנחנו יודעים שלא כך המצב. אין סיבה לתת לצומת את ה-value הכי נמוך, אלא ניתן להעריך אותה כתוחלת הערכים האפשריים.

* + - נצפה שבממוצע מספר הפעמים ש- סוכן Expectimax יפסיד יהיה גדול יותר ממספר הפעמים שסוכני *minimax* או *alpha-beta* יפסידו.
    - נצפה שבמצב שבו פקמן לכוד בין רוחות נראה התנהגות שונה בין אלגוריתם זה לאלגוריתמים הקודמים. עבור האלגוריתמים הקודמים, ההנחה היא תמיד שהרוחות יעשו את הצעד הכי גרוע לפקמן, פקמן יחשוב שאין לו מצב שלא מוביל להפסד ולכן יפסיד בודאות. מה שקורה בפועל הוא שיש סבירות מסוימת שהרוחות ילכו לכיוון שלא מוביל לפקמן, כלומר לא הכי גרוע בשבילו. סוכן Expectimax ידע להתחשב באפשרות זו ולכן חלק מהפעמים יחלץ מהמלכוד וינצח.

1. **בניית סוכן Expectimax לרוח לא רנדומלית**
2. **ניסוח השערות במשחק פאקמן**
3. **ניסויים תוצאות ומסקנות**
4. **תחרות בקורס**