חלק א' – הכרות עם המשחק

**תרגיל 3 – ReflexPlayer**

שחקן הבסיס מקבל את מצב המשחק הנוכחי וראשית מחשב את כל הפעולות האפשריות (החוקיות) עבורו בהינתן מצב המשחק. לאחר מכן, עבור כל פעולה אפשרית מפתח את המצב העוקב שהיא יוצרת ועבורו מחשב את הניקוד המשחק שישיג אם יבצע מהלך זה, אוסף את הניקוד של כל המצבים האפשריים (עבור כל הפעולות האפשריות) ומוצא את הניקוד המקסימלי. לבסוף בוחר רנדומלית פעולה אשר תבטיח כי המצב שאליו יגיע השחקן יניב עבור את הניקוד המקסימלי שחושב. כלומר, האלגוריתם שבו משתמש השחקן הינו Greedy Best 1st ומעריך מצבים על ידי הניקוד שהם מניבים.

חלק ב' – בניית סוכן משופר

**תרגיל 1 – Better Evaluation Function**

הגדרת הפונקציה היוריסטית שלנו:

.

**תרגיל 2 – Motivation:**

נשים לב כי היוריסטיקה שלנו מעריכה את המצב ראשית לפי הניקוד הכללי במשחק שבו הפאקמן מקבל במצב זה בדיוק כמו היוריסטיקה המקורית.  
בנוסף היוריסטיקה שלנו מתחשבת במיקום הרוחות על הלוח ומעריכה מצבים לפי המרחק (מרחק מנהטן) המינימלי (המקרה הגרוע ביותר, ככל שקרוב יותר לרוח קרוב לפסילה) של הפאקמן מרוח כלשהי ומוסיפה מספר זה אל הערך היוריסטי של המצב ומתעדפת מצבים שמתרחקים מהרוח הקרובה אל הפאקמן.  
על מנת לצבור יותר נקודות הפונקציה מעריכה את המצב גם לפי המרחק (מרחק מנהטן) של המצב מנקודת על המפה אשר מכילות אוכל, ככל שמרחק זה יותר קטן הערך שמוחסר מהערך של המצב הינו קטן יותר והערך היוריסטי הסופי שיוחזר גדול יותר כלומר מעדיפה מצבים שקרובים יותר לאוכל.  
המוטיבציה מאחורי היוריסטיקה היא חתת ניקוד גבוה יותר למצבים שרחוקים מרוחות וקרובים לאוכל וממקסמים את הניקוד הכללי.

חלק ג' – בניית סוכן Min – Max

**תרגיל 1 – הנחה**

בבניית עץ Min-Max אנו מניחים כי המשחק הינו משחק תורות והסוכנים האחרים, כלומר הרוחות, זזים כל אחד בתורו אמנם במשחק זה כל הסוכנים מבצעים מהלך ביחד ואין מדובר במשחק תורות.

**תרגיל 3 – מימוש נוסף**

במקום לממש שכבה עבור כל סוכן שמשתתף, כלומר עבור כל רוח, היינו מייצרים שכבה אחת לכל הרוחות כאשר בשכבה זו כל מצב היה מגלם את כל הפרמוטציות האפשרויות של מהלכים חוקיים עבור כל הסוכנים האחרים. לדוגמא בהינתן שיש 2 רוחות במשחק ולכל אחת יש 4 מהלים חוקיים לבצע, כלומר תזוזה לכל ארבעת הכיוונים נקבל 16 קשתות שיוצאת מהמצב של הרוחות , אשר כל קשת מסמלת את המהלך של כל רוח במשחק:

(north, north), (north, south) ……………… (east, east)

יתרונות השיטה החדשה: בצורה זו נקבל עץ יותר שטוח כי כעת אנו דוחסים את שכבות הסוכנים היריבים אל שכבה אחת וכך עומק יותר קטן נוכל לחשוף יותר מהלכים עתידיים.

חסרונות השיטה החדשה: מקדם הסיעוף בעץ עבור השיטה החדשה הוא אקספוננציאלי בכמות הסוכנים המשתתפים במשחק כלומר אם יש n רוחות במשחק מקדם הסיעוף יכול גם להגיע ל 4n-1 דבר שלא יעיל לדוגמא עבור האלגוריתם אלפא – ביתא אשר מספר הפיתוחים תלוי אקספוננציאלית במקדם הסיעוף.

חלק ד' – בניית סוכן Alpha – Beta

**תרגיל 1 – מבנה העץ**

מבנה העץ שהוגדר הוא בעל מקדם סיעוף שחסום על ידי 4, לכל אחד מהסוכנים יש במקרה הכי טוב מהלך לכל אחד מארבעת הכיוונים. ולכן בממוצע אלגוריתם אלפא – ביתא מבצע: פיתוחים. לעומת זאת המבנה שאנחנו הצענו יש לכאורה 2 סוכנים, הפאקמן ושאר הרוחות לכן במקרה בממוצע יבצע  כלומר יותר פיתוחים.  
מסקנה: מבנה העץ שהוגדר טוב יותר עבור אלגוריתם אלפא-ביתא כאשר מספר הסוכנים הוא גבוה כי מקדם הסיעוך בו נשאר קבוע.

**תרגיל 3 – השוואה**

מבחינת זמן ריצה לסוכן alpha beta ייקח פחות זמן לחשב את הצעד הבא לביצוע ולכן זמן הריצה שלנו יקטן, ניתן גם לראות זאת על ידי הגדלת העומק לחישוב ל -4 ולראות כי לסוכן זה לוקח פחות זמן לחשב את הצעד הבא לביצוע (המשחק פחות נתקע) ולכן זמן הריצה יקטן.  
מבחינת בחירת המהלכים האלגוריתמים הם זהים, זאת מכיוון שאלגוריתם שalpha-beta גוזם רק ענפים שלא ישנו את ערך האסטרטגיה הסופי ולכן יוחזר אותו מהלך לביצוע בעל הערך הגדול ביותר.