חלק א'

3. ראשית כל השחקן הבסיסי ReflexPlayer מייצר רשימה של פעולות חוקיות מהמצב הנוכחי במשחק. עבור כל פעולה ברשימה, השחקן מחשב את התוצאה של פונקציית ההערכה על מצב המשחק לו היה נוקט בפעולה זו. מבין כל התוצאות הנ"ל, השחקן מוצא את התוצאה בעלת הערך המקסימלי. ייתכן וערך זה מתאים למספר פעולות ועל כן השחקן מגריל מבין התוצאות בעלות הערך המקסימלי פעולה כלשהי ומבצע אותה.
ההיוריסטיקה שבה השחקן הבסיסי משתמש היא תוצאת המצב הנוכחי (כפי שמוצגת בממשק הגרפי).

חלק ב'

נגדיר את הקבועים הבאים:

FOOD_FACTOR =10
MOBILITY_FACTOR = 1
CAPSULE_FACTOR = 200
SCARED_GHOST_FACTOR = 200
NOT_SCARED_FACTOR = 50
CAPSULE_NUM_FACTOR = 2000
WINDOW = 4

1. נגדיר תחילה את הפרמטרים שישמשו אותנו לחישוב ההיורסטיקה החדשה:

- <u>ניידות:</u> נגדיר ניידות כמספר הצעדים האפשריים שיש לפקמן לבצע מהמצב הנוכחי. מספר המצבים הנ"ל הוא בתחום בין $possible\ steps \in \{0,1,2,3,4\}$.

$$mobilityFactor = MOBILITY\_FACTOR \cdot number\ of\ legal\ actions$$

- <u>קרבה לאוכל:</u> נחשב מרחק מאוכל כמרחק מנהטן מהמיקום של פקמן לפיסת אוכל. לצורך חישוב ההיוריסטיקה נרצה להתייחס למרחק מפיסת האוכל הקרובה ביותר, נסמנה $closest\_food\_dist$. כלומר, במהלך המשחק ניתן ערך גבוה יותר לאוכל קרוב יותר וערך נמוך יותר לאוכל רחוק יותר. Closer food = better state.

$$foodFactor = FOOD\_FACTOR \cdot \left(\frac{1}{closest\_food\_dist}\right)$$

- <u>קרבה לכמוסה:</u> באופן דומה לפרמטר הקודם, נרצה לחשב גם את מרחקו של פקמן מן הכמוסה הקרובה ביותר אליו כאשר המרחק מחושב לפי מרחק מנהטן. נסמן מרחק זה ב-$closest\_capsule\_dist$.

$$capsuleDistFactor = \begin{cases} CAPSULE\_FACOR \cdot \left(\frac{1}{closest\_capsule\_dist}\right), & \text{number of capsules on board} > 0 \\ 0, & \text{else} \end{cases}$$

- <u>מספר הכמוסות:</u> כפי שהשם מרמז, נרצה להכניס לשיקול בחישוב ההיוריסטיקה את כמות הכמוסות שנמצאות על הלוח. פרמטר זה דומה לפרמטר הקודם במובן ששניהם אמורים לסייע לפקמן לאכול את הכמוסות אך נדרשת הפרדה של הפרמטרים בכדי להגיע לחישוב מדויק יותר.

$$capsuleNumFactor = \begin{cases} CAPSULE\_NUM\_FACOR \cdot \left(\frac{1}{number\_of\_capsules}\right), & \text{number of capsules on board} > 0 \\ 2 * CAPSULE\_NUM\_FACOR, & \text{else} \end{cases}$$

- <u>קרבה לרוח:</u> למעשה, נרצה להפריד למקרים- מרחק מרוח אשר מפחדת אל מול רוח אשר אינה מפחדת. נבחן את המרחק מינימלי (מחושב ע"י מרחק מנהטן) לרוח מבין כל הרוחות המפחדות וכן את המרחק המינימלי לרוח מבין הרוחות שאינן מפחדות. נסמן את המרחקים ב- min_scared_ghost_dist ו-min_not_scared_ghost_dist בהתאמה. נפריד כעת למקרים:

רוח מפחדת:
הנוסחה עבור חישוב קרבה לרוח שמפחדת:

$$scaredFactor = \begin{cases} \text{SCARED\_GHOST\_FACTOR} \cdot \left(\frac{1}{\text{min\_scared\_ghost\_dist}}\right), & \text{min\_scared\_ghost\_dist} > 0 \\ 0, & \text{else} \end{cases}$$

עבור רוח שאינה מפחדת:
נגדיר "חלון" –התחום שסובב את פקמן כמרחק אשר קטן או שווה ל-4 ממנו על פי מרחק מנהטן.
כעת נגדיר את הנוסחה עבור חישוב קרבה לרוח שאינה מפחדת:

$$notScaredFactor = \begin{cases} \text{NOT\_SCARED\_FACTOR} \cdot WINDOW & \text{min\_not\_scared\_ghost\_dist} > \text{WINDOW} \\ \text{NOT\_SCARED\_FACTOR} \cdot \text{min\_not\_scared\_ghost\_dist} & else \end{cases}$$

- <u>תוצאת המשחק:</u> נרצה להשתמש בתוצאת המשחק של כל מצב (כפי שמוצגת בממשק הגרפי) אשר שימשה לחישוב ההיוריסטיקה הבסיסית הנתונה בתרגיל. נסמנה $gameScore$.

לבסוף, נגדיר את ההיוריסטיקה להיות:

$$\text{EvaluationFunction}(state) = 5 \cdot gameScore + foodFactor + mobilityFactor + capsuleDistFactor + capsuleNumFactor + scaredFactor + notScaredFactor$$

2. <u>מוטיבציה:</u>

- <u>ניידות –</u>
ככל שיהיו לפקמן יותר אפשרויות לבחור מהם, כך יכולתו למקסם את הניקוד שלו תגדל. בנוסף לכך, ניידות גדולה יותר תאפשר לפקמן אפשרויות בריחה רבות יותר בעת הצורך בבריחה מרוח.

- <u>קרבה לאוכל –</u>
ככל שנאכל יותר אוכל כך פקמן יצבור יותר נקודות, והניקוד הכולל עבור המשחק יהיה גבוה יותר. בנוסף אכילת כלל המאכלים על הלוח תוביל את פקמן לניצחון.

- <u>קרבה לכמוסה –</u>
אכילת כמוסות מאפשרת לפקמן לאכול רוחות. אכילת רוחות במשחק היא הפעולה הרווחית ביותר, ולכן על ידי אכילה של הכמוסות ולאחר מכן אכילת רוחות, נוכל למקסם את ציונו של פקמן.

- <u>מספר הכמוסות –</u>
פרמטר זה בא לענות על אותם צרכים שהוגדרו בקרבה לכמוסה. הסיבה להוספתו היא שבעת אכילת כמוסה המרחק הקרוב ביותר הבא שיחושב עבור אכילת כמוסה יהיה גדול מהמרחק שמחושב כרגע ביחס לכמוסה אותה אנו עומדים לאכול. לפיכך, נרצה להכניס פרמטר נוסף בעל משקל גדול יותר שיורה לפקמן לאכול את הכמוסות במשחק.

- <u>קרבה לרוח –</u>
נרצה להפריד למספר מקרים:
  - אם הרוח מפחדת– נרצה לאכול את הרוח וכך להרוויח ניקוד גבוה יותר.
  - אם הרוח אינה מפחדת ופקמן קרוב אליה – נרצה שפקמן יברח ממנה על מנת להישאר בחיים.
  - אם הרוח אינה מפחדת ופקמן רחוק ממנה – נרצה להזניח את הפרמטר כדי שפקמן לא יהיה במצב של פחד תמידי.

- <u>תוצאת המשחק –</u>
כמו בהיוריסטיקה הנתונה בתרגיל, נרצה גם אנחנו להכניס לשיקול את תוצאת המשחק המתקבלת בכל מצב, שכן תוצאה זו משקללת פרמטרים נוספים כגון: ניקוד על ניצחון, ניקוד על אכילת רוח וכו'.

- <u>הסבר למשקל הקבועים –</u>
נשים לב ליחס של 1:4 בין רוח שמפחדת לרוח שאינה מפחדת. הבדל זה נועד לגרום לפקמן לברוח מרוח שאינה מפחדת ולתת לזה משקל רב יותר בהיוריסטיקה, כי יותר חשוב לנו לא למות מאשר להרוויח עוד נקודות.
קבוע מספר הכמוסות הוא מאוד גדול, שכן אנחנו רוצים שפקמן יצרוך את כלל הכמוסות שיש במשחק וכך יגדיל את סיכויו לאכול את הרוחות ולמקסם את ציונו.
יחס האוכל אל מול רוח שמפחדת, העדיפות היא ללכת לאכול רוח שמפחדת אשר קרובה לפקמן, שכן אכילת רוח נותנת יותר נקודות מאשר אכילת אוכל.
מלבד הנקודות הללו בחירת ערך לקבועים נעשתה באופן שרירותי על ידי ניסוי וטעיה.

<u>חלק ג'</u>

1. בעת שימוש בעץ ה-$MiniMax$ המתואר ישנה הנחה סמויה כי המשחק מנוהל בתורות. כלומר, ישנו סדר דטרמיניסטי שבו הסוכנים מקבלים החלטות במהלך המשחק. סדר זה מתבטא כך שכל שכבה בעץ מייצגת סוכן אשר מקבל את החלטתו על סמך ההחלטות שמקבל הסוכן שמייצג את השכבה מתחתיו בעץ. הנחה זו אינה בהכרח נכונה שכן לעומת משחקים כדוגמת שחמט, שבהם אכן יש משמעות לתור וכל סוכן "מגיב" לצעד קודם בזמן שקרה ע"י סוכן אחר (אחד או יותר), בפקמן אין כך הדבר. בפקמן כל ההחלטות של כלל הסוכנים מתבצעות במקביל ואין לסוכנים ידע קודם על הצעד שהולך להתבצע במקביל לצעד שלהם.
הנחה נוספת בשימוש בעץ היא שכל הרוחות חולקות את אותה אסטרטגיה והיא לגרום לפקמן להפסיד. גם הנחה זו אינה בהכרח נכונה, והיא תלויה בסוג הרוח. ישנן רוחות אשר נעות באקראיות על המסך ואין להן מטרה להגיע לפקמן.

3. שיטה נוספת למימוש אלגוריתם $MiniMax$ שבה לא מייצרים שכבה בעץ עבור כל רוח תוכל להשתמש בשכבת מינימום יחידה המשקללת את הצעדים של כלל הרוחות. נניח כי במשחק

ישנן n רוחות. כעת, כל קשת היוצאת מצומת מינימום לא תייצג את הפעולה שתנקוט רוח יחידה אלא וקטור בגודל n כך שבכניסה ה-i מופיע הפעולה של הרוח ה-i. לכן, מספר הקשתות שיצאו מצומת מינימום יהיו מספר האפשרויות הכולל של צירופי צעדי הרוחות השונות. (למשל: אם $n = 2$, לרוח הראשונה יש 2 צעדים אפשרים ולרוח השניה 3 צעדים נקבל כי מצומת המינימום שמייצג את כל הרוחות יוצאות $2 \cdot 3 = 6$ קשתות).
נרצה להשוות בין שני המימושים הן מבחינת סיבוכיות החישוב והן מבחינת הביצועים.

סיבוכיות:

נבחן את מספר הצמתים בעץ ה-$MiniMax$ ב"סבב תורות" יחיד- כלומר משכבת מקסימום אחת ועד שכבת המקסימום הבאה. ברור כי שכבת המקסימום זהה בשני המימושים ולכן מספר צמתי המקסימום זהה ונוכל לבחון את מספר הצמתים המפותחים מצומת מקסימום יחיד. צומת זה מייצג את מספר הצעדים החוקיים של פקמן ונסמן את מספר הבנים שלו ב-m_{pacman}. נניח כי במשחק ישנן $1,2,\dots,n$ רוחות כאשר מספר הצעדים האפשריים של הרוח ה-i הינו m_i וסדר הרוחות תואם את סדר השכבות בעץ (=שכבת המינימום העליונה ביותר היא של רוח מספר 1 וכן הלאה.)

<u>אבחנה:</u> מספר ההסתעפויות מכל צומת מינימום\מקסימום באותה שכבה שווה במשחק פקמן. הסיבה לכך היא שכל קשת שיוצאת מצומת מייצגת צעד חוקי של הסוכן והיא תלויה בלוח בלבד ולא בסוכנים אחרים של המשחק.

במימוש המוצע בתרגיל (מספר שכבות מינימום כמספר הרוחות) נוכל לומר כי מספר הצמתים בעץ (תוך התעלמות מצמתי מקסימום) הינו:

$m_{pacman} + m_{pacman} \cdot (m_1 + m_1 \cdot m_2 + m_1 \cdot m_2 \cdot m_3 + \dots + (m_1 \cdot \dots \cdot m_n))$ כאשר ההצדקה לחישוב היא האבחנה לעיל.

לעומת זאת, במימוש המוצע בסעיף ומשתמש בשכבת מינימום יחידה מספר הצמתים בעץ הינו:

$m_{pacman} \cdot m_1 \cdot m_2 \cdot \dots \cdot m_n$.

קל לראות כי מספר הצמתים בשכבת מינימום יחידה מוכל ממש במספר הצמתים עבור המימוש המקורי. באופן דומה, גם מספר הרצות $Succ$ במימוש המקורי גדול ממש מאשר במימוש החדש ועל כן סביר שהמימוש החדש יהיה מהיר יותר.

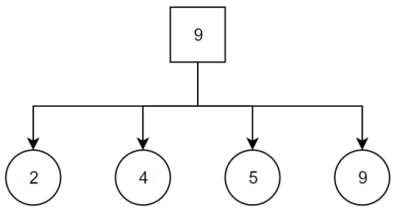
ביצועים:

נשים לב שהמימוש החדש לא מניח את ההנחה הראשונה שהוצגה בסעיף 1 בצורתה המקורית (לגבי קיום סדר שבו הסוכנים מקבלים החלטות במהלך המשחק). כעת, ההנחה המתאימה היא שיש סדר לא בין כלל הסוכנים אלא רק בין פקמן והרוחות ככלל. אפשר להסתכל על כלל הרוחות כעל סוכן יחיד אשר משחק בתורות אל מול פקמן. מסיבה זו, קשה להשוות בין הביצועים הנשענים על הנחות עבודה שונות.

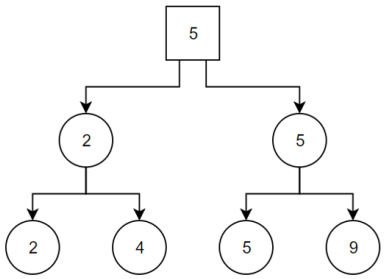
נראה באמצעות דוגמה (1) כי ערכי המקסימום במימוש המקורי חסומים מלמעלה ע"י ערכי המקסימום בעץ מינימיקס במימוש החדש. (ניתן להוכיח את הטענה במלואה אבל זו לא מהות השאלה אז נזניח את ההוכחה). כלומר, נקבל כי במימוש החדש תוצאות המינימקס יניבו ערכי מקסימום אופטימיים יותר מאשר במימוש המקורי ולכן אולי פקמן ינהג בחוסר זהירות. אמנם, נוכל לצפות כי המימוש החדש יניב תוצאות טובות, כי הוא ממדל בצורה המטיבה יותר עם המציאות. הוא ממדל בצורה טובה יותר כיוון שכל הרוחות פועלות במקביל בעוד שבמימוש הראשון, הרוחות פועלות בתורות.

דוגמה (1):

נניח כי יש 2 רוחות, לכל אחת 2 צעדים אפשריים, והשכבה התחתונה קיבלה את ערכיה מהעלים. עיגול מסמל צומת בשכבת מינימום וריבוע מסמל צומת בשכבת מקסימום.



מימוש חדש



מימוש מקורי

חלק ד':

1. מבנה העץ החדש שהוגדר לא משפיע על אלגוריתם אלפא-בטא.

כפי שנלמד בהרצאה, אלגוריתם אלפא-בטא הינה טכניקה המאפשרת לחיפוש מינימקס, תוך פעפוע חסמים, לגזום חלק מענפי החיפוש תוך שמירה על אופטימליות האלגוריתם. האלגוריתם מכניס לשיקול את חסמי אלפא ובטא כאשר חסם אלפא מוגדר כערך עכשווי מקסימלי מבין כל אבות ה-$\mathrm{M}ax$ הקדמונים וחסם בטא מוגדר כערך עכשווי מינימלי מבין כל אבות ה-Min **הקדמונים**. בפרט, אלגוריתם אלפא-בטא מוגדר ביחס לשכבה בעץ ולא ביחס למספר שכבות המינימום והמקסימום בעץ. כלומר, חסם בטא יחושב ביחס לכל כל אבות ה-Min הקדמונים, בלי חשיבות לכך שבעץ הופיעו מספר שכבות מינימום עוקבות ולכן הנכונות והאופטימליות של האלגוריתם לא יושפעו.

למרות האמור לעיל, וההבטחה על נכונות התוצאה של האלגוריתם, ייתכנו מצבים שבהם ערכי הביניים בשכבות המינימום אינם מעודכנים נכונה. כלומר, ייתכן שנגזום ענפים אשר היו משפיעים על ערכי צמתי ביניים (ערכי צמתים בשכבות מינימום שאינן השכבה הנמוכה ביותר). יחד עם זאת, ההסבר לעיל נשאר תקף ועל כן תוצאת המינימקס שתתקבל לא תושפע מגזימות אלו.

3.

a. נצפה שסוכן $AlphaBetaAgent$ יהיה מהיר יותר מסוכן $MinMaxAgent$.

הסיבה לכך היא שמטרת אלגוריתם אלפא-בטא היא קיצוץ ענפים בעץ שפיתוחם בהכרח לא ישנה את ערך המינימקס. כלומר, האלגוריתם חוסך פיתוח של צמתים שלא תורמים מידע חדש לאסטרטגיה וכמובן שנצפה שהרצה שבה מפתחים את כל הצמתים האפשריים תהיה איטית יותר מאשר הרצה שבה מפתחים מספר צמתים קטן יותר. תיתכן הרצה שבה סוכן $AlphaBetaAgent$ יפעל באותה מהירות כמו סוכן $MinMaxAgent$ אך נצפה כי לאורך משחק שלם ניתקל במספר הזדמנויות לבצע גיזום אלפא-בטא.

b. נצפה שסוכן $AlphaBetaAgent$ יבחר בדיוק באותם מהלכים כמו סוכן $MinMaxAgent$. ממשפט נכונות אלפא-בטא האלגוריתם מחזיר את ערך המינימקס של העץ. כלומר, בכל הפעלה של אלפא-בטא יתקבל בדיוק אותו הערך שמתקבל מהפעלה של אלגוריתם מינימקס (תחת אותו מצב של המשחק, סוכן ועומק עץ לחיפוש). בפרט זה נכון עבור ההפעלה הראשונית של שני הסוכנים המתאימים. כיוון שההרצה הראשונית תניב את אותו צעד לפי משפט הנכונות אזי מצב המשחק החדש יהיה זהה אצל שני הסוכנים ובאינדוקציה גם המהלכים הבאים ייבחרו באופן זהה אצל שני הסוכנים.

חלק ה':

2. השינוי ביחס לסוכנים המשתמשים באסטרטגיית MinMax היא שסוכן Expectimax לרוח רנדומלית אינו מנסה לגרום לפקמן להפסיד אלא נע בצורה אקראית על המסך. לעומתו, באסטרטגיית MinMax הנחנו תמיד כי היריבים (=הרוחות) יפעלו בצורה אנוכית. כלומר- כל רוח תרצה לנצח, או באופן שקול: כל רוח תרצה שפקמן ייפסל.

נצפה כי תוצאות ההרצה יהיו עם ניקוד גבוה מאשר משחקים שהסתמכו על אסטרטגיית MiniMax שכן ההרצות מתבצעות עם הרוח הרנדומלית אשר בוחרת באקראי ובאופן אחיד באיזה פעולה לנקוט ביחס לסט הפעולות האפשריות. על כן, סוכן Expectimax יהיה נאמן יותר למציאות במקרה זה בעוד סוכן MiniMax יניח הנחות פסימיות שהסיכוי שלהן להתממש קטן.

לדוגמא, נניח כי פקמן כלוא בין שתי רוחות וכל צעד שיחליט לעשות יקדם אותו אל עבר אחת מהן וירחיק אותו מהשנייה. נניח בנוסף כי אחת מהן תחליט לפנות לכיוונו כי זה הצעד החוקי היחיד בעבורה בעוד השניה תחליט ללכת לכיוון שונה. אם פקמן ישתמש באלגוריתם מינימקס הוא יניח את הגרוע מכל ויחליט ללכת באקראי לכיוון אחת מהן (נניח כי הן במרחק שווה ואין עוד אלמנטים שמשפיעים על החלטתו) ולכן בסיכוי 0.5 ילך לרוח שפונה לכיוונו ויגדיל את סיכויו למות. לעומת זאת, בשימוש בסוכן Expectimax, פקמן יראה כי לאחת הרוחות סיכוי נמוך יותר ללכת בעקבותיו ולכן יחליט ללכת אחריה ולא יבטיח לעצמו מוות בעתיד הקרוב.

חלק ו':

1. התפלגות התנועה של הרוח מחושבת על ידי שקלול של הסתברויות שמכניסות לשיקול באיזו הסתברות הרוח מתכוונת לתקוף (אם היא לא מפחדת) או, במקרה שהיא כן מפחדת באיזו הסתברות הרוח מתכוונת לברוח. במקרה שהרוח מתכוונת לתקוף, היא תרצה לבצע מהלך מבין המהלכים שיניבו את התוצאה הטובה ביותר עבורה ותבחר מביניהם בהתפלגות יוניפורמית. במידה והרוח לא מתכוונת לתקוף היא תבחר בהתפלגות יוניפורמית מבין כלל המהלכים. באותה צורה הרוח מחשבת עבור הכוונה לברוח במצב של פחד. נפרט כעת את החישוב במלואו ובסופו נצרף את הנוסחה לחישוב ההסתברות לבחירת כל צעד.

האסטרגטיה המלאה כוללת חישוב של ההסתברות לבחירת כל צעד התקדמות אפשרי של הרוח. ראשית, נרצה לחשב את התוצאה הטובה ביותר (נסמנה $bestScore$) שהיא המרחק המינימלי/מקסימלי מפקמן בהתאם להאם הרוח מפחדת או לא (עבור רוח שאינה מפחדת התוצאה המקסימלית תהיה המרחק המינימלי מפקמן מבין הצעדים האפשריים, ולהפך). נסמן בנוסף את המהלכים שלאחר ביצועם תימצא הרוח במרחק $bestScore$ ב-$bestActions$.
כמו כן, מוגדר פרמטר $bestProb$ שמייצג את ההסתברות שהרוח תרצה לתקוף/לברוח. כעת כשבידינו שני הפרמטרים הללו נוכל לחשב את ההסתברות לבחירת כל צעד:

$$P(take\ action\ x) = I \cdot bestProb \cdot \left(\frac{1}{|bestActions|}\right) + (1 - bestProb) \cdot \left(\frac{1}{|legalAcions|}\right)$$

כאשר I הוא אינדיקטור שמקבל ערך 1 אמ"מ $x \in bestAcions$.

כלומר, נשים לב שכל המהלכים מקבלים ראשית את אותה ההסתברות- זו שהייתה מתקבלת לו הרוח הייתה רוחה רנדומלית. הסתברות זו מקבלת פקטור בהתאם להסתברות שהרוח לא תרצה לתקוף/לברוח שכן במקרה זה הרוח היא אכן רנדומלית. אבל עבור $Directional\ Ghost$ מכניסים לשיקול גם את האופציה שהרוח כן רוצה לתקוף/לברוח ובמקרה זה ההסתברות של צעד תקבל תוספת להסתברות הבסיסית אם זה מהלך אופטימלי. לבסוף כלל ההסתברויות של כל הצעדים האפשריים מנורמלות כך שיסתכמו ל-1.

בנוסף, נשים לב כי לפי הקוד הרוח המפחדת היא איטית פי 2 מאשר רוח במצב הרגיל.

3. ההבדל במימוש בין DirectionalExpectimaxAgent ל RandomExpectimaxAgent נעוץ בחישוב התוחלת של המצבים ההסתברותיים. למעשה, בשני המקרים חישוב התוחלת מתבצע עפ"י הנוסחה הבאה:

$$Expected\ value = \sum_{action \in Legal\ actions} (result\ of\ expectimax\ if\ we\ choose\ action) \cdot (probability\ the\ we\ choose\ action)$$

ההבדל נובע מחישוב ההסתברות לבחירת כל מצב $(probability\ the\ we\ choose\ action)$.
חישוב ההסתברות ב – RandomExpectimaxAgent מחושב על פי הנוסחה:

$$probability\ the\ we\ choose\ action = \frac{1}{|Legal\ actions|}$$

בעוד חישוב ההסתברות ב- DirectionalExpectimaxAgent מחושב על פי הנוסחה שהוצגה בסעיף 1 בחלק זה.

4. נציע מספר רעיונות לשיפור אסטרטגיית הרוחות:

a. "הפרד ומשול" – נחלק את הלוח למספר חלקים כמספר הרוחות אשר משתתפות במשחק. כל רוח תמומש באמצעות DirectionalGhost ותהא אחראית לאזור תחום בלוח המשחק. אסטרטגיה זו תשפר את הרוחות מכיוון שאנו מחלקים את הלוח למספר חלקים, וכך לא "נבזבז" משאבים של מספר רוחות במקום קטן. כתוצאה מכך, נצפה שההסתברות של פקמן לפגוש רוח תגדל.

b. "שטח השמדה" – הרוח תפעל עם אסטרטגיית DirectionalGhost למעט מניפולציה על הסתברות התקיפה (prob_attack). נגדיר ערך סף, כך שאם מרחק מנהטן בין פקמן והרוח קטן מערך זה, אזי הסתברות התקיפה תעודכן ל-1, אחרת נפעל כרגיל.
אסטרטגיה זו תשפר את הרוחות מכיוון שעדכון ערך התקיפה מבטל את הצעדים האקראיים שלא מטיבים עם הרוח בקרבת פקמן. כתוצאה מכך, הרוח תדע לנצל הזדמנויות לאכול את פקמן כאשר הוא בקרבתה בצורה "חכמה" יותר מאשר DirectionalGhost.

c. "דגירה" – כל רוח תבחר ל"דגור" על הכמוסה הקרובה ביותר אליה. כלומר הרוח תתקרב עד כדי מרחק מינימילי מהכמוסה, ותנוע הלוך וחזור בקרבתה. בצורה זו, נמנע מפקמן בהסתברות גבוהה את האכילה של הכמוסה. בעקבות זאת, בהסתברות גבוהה (כתלות במספר הרוחות והכמוסות שעל הלוח) פקמן לא יוכל לאכול את הרוחות ואף לא את האוכל שבקרבתן, ולכן לא יוכל לנצח.

d. "מארב ארעי" – הרוח תפעל עם אסטרטגיה הדומה ל- DirectionalGhost כאשר כעת חישובי ההסתברויות ישקללו בנוסף את המרחק מיתר הרוחות. חישוב ההסתברויות יהיה כזה כך שמרחק קרוב יותר לרוח אחרת יוביל להסתברות קטנה יותר לבחירת הפעולה. בצורה כזו נייצר מצב שבו הרוחות השונות מנסות כולן לתקוף את פקמן אך להגיע אליו מכיוונים שונים. אסטרטגיה זו תשפר את הרוחות מכיוון שהיא תגדיל את ההסתברות לאכילת פקמן (כי יהיו לו פחות דרכי מילוט).

חלק ז'

סיטואציה: נרצה לקבוע אם במשחק פקמן המשתמש בסוכן $MinimaxAgent$ פקמן יגיע לתוצאה גבוהה יותר כאשר הוא משחק מול DirectionalGhost או כאשר הוא משחק מול RandomGhost.

ננסח את השערת האפס - H_0:
במשחק המשתמש בסוכן $MinimaxAgent$, פקמן יגיע לתוצאה גבוהה יותר כאשר הוא משחק מול RandomGhost מאשר מול DirectionalGhost.

ננסח את ההשערה החלופית- H_1:
במשחק המשתמש בסוכן $MinimaxAgent$, פקמן יגיע לתוצאה זהה בין אם הוא משחק מול RandomGhost או מול DirectionalGhost.

כדי לבחון את ההשערות נערוך את הניסוי הבא:
נבצע 100 סבבים כאשר בכל סבב נריץ משחק אחד של סוכן $MinimaxAgent$ מול DirectionalGhost ומשחק אחד של סוכן $MinimaxAgent$ מול RandomGhost. כמובן שהמשחקים מתנהלים תחת אותם תנאים בהקשר של עומק החיפוש, מספר הרוחות, לוח המשחק וכו'. בנוסף, נגדיר כי הבחירה הרנדומית מבין צעדים בערכים שווים תתבצע בצורה דטרמיניסטית עם seed קבוע עבור כל המשחקים.
לכל סבב יוגדר "ניצחון" אם RandomGhost "מנצח" את DirectionalGhost.
ניצחון של RandomGhost על DirectionalGhost מוגדר באופן הבא:
אם התוצאה של פקמן במשחק מול RandomGhost גבוהה מאשר מול DirectionalGhost אזי נאמר כי RandomGhost מנצח.
אם התוצאה של פקמן במשחק מול RandomGhost נמוכה מאשר מול DirectionalGhost אזי נאמר כי RandomGhost מפסיד.
אם התוצאה של פקמן במשחק מול RandomGhost שווה לזו שהתקבלה מול DirectionalGhost אזי נטיל מטבע הוגן ונאמר כי RandomGhost מנצח אם יצא "עץ".

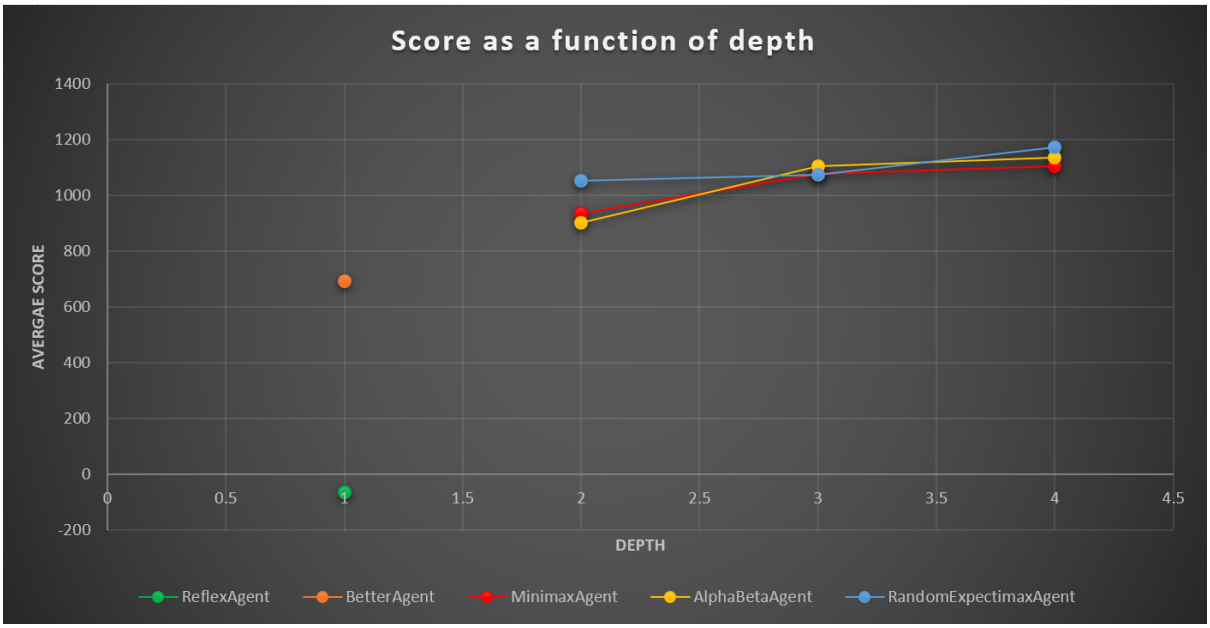
$$X_1, \dots, X_{100} \sim Ber(p)$$
$$X = \sum_{i=1}^{100} X_i \sim Bin(100, p)$$
$$H_0:\ p > 0.5 \qquad H_1:\ p = 0.5$$

נגדיר את המבחן הבינומי (=כלל החלטה) הבא:
נדחה את השערת האפס אם מספר הניצחונות (RandomGhost "מנצח" את DirectionalGhost) בהרצת 100 משחקים קטן או שווה מ-10.
לא נדחה את השערת האפס אם מספר הניצחונות גדול מ-10.

חלק ח':

1. נציין כי בוצעו 7 הרצות לכל סט תנאים, כנדרש בשאלה.

2. I.



II.

d=1	d=2	d=3	d=4	
-63.1286				ReflexAgent
693.3857				BetterAgent
	934.6571	1077.914	1105.7	MinimaxAgent
	901.6	1104.185714	1134.429	AlphaBetaAgent
	1052.3	1073.357	1171.186	RandomExpectimaxAgent

3. מהגרף והטבלה ניתן להסיק מספר מסקנות:

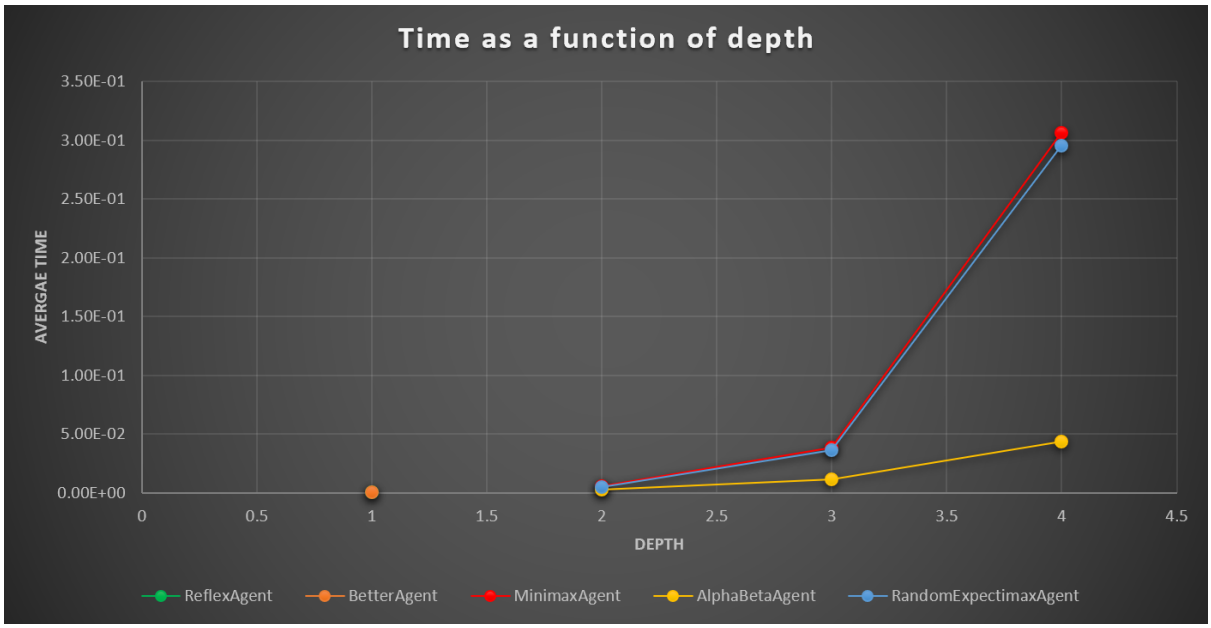
- ישנו שיפור משמעותי בין ההיוריסטיקה הנתונה לבין היוריסטיקה בעלת הפרמטרים אשר הגדרנו בסעיף הראשון. זה אכן תואם את ציפיותינו שכן ההיוריסטיקה שבה משתמש ReflexAgent היא יחסית מנוונת ולא מכניסה לשיקול פרמטרים רבים על מצב הלוח (פרמטרים אלו מפורטים בחלק ב' סעיף 1).
- עבור הסוכנים שיודעים להסתכל לעומק, ניתן לראות קורלציה בין מספר הצעדים שמסתכלים קדימה לבין התוצאה הסופית. ככל שמעמיקים יותר, כך התוצאות שיתקבלו טובות יותר (למעט במעבר בין עומק 3 ל-4 עבור AlphaBetaAgent אצלה ההבדל בתוצאות במקרה זה הוא יחסית קטן). אבחנה זו עונה על ציפיותנו שכן ככל שפקמן מחשב עומק גדול יותר כך הוא מסתכל יותר צעדים קדימה ויודע לקבל החלטה מושכלת יותר בנוגע לצעד הקרוב שעליו לבצע ולכן זה משתלם לו לטווח הארוך.
- נשים לב שעבור MinimaxAgent ו-AlphaBetaAgent השיפור בתוצאה במעבר בין עומק 2 לעומק 3 הוא גדול מאשר במעבר בין עומק 3 לעומק 4. נוכל להסיק מכך שקיים עומק מסויים עבורם אשר ממנו נהיה חסומים על ידי חסם אסימפטוטי, שכן לא נוכל לשפר יותר את התוצאות בהסתמך על אותה ההיוריסטיקה. הציפייה שלנו הינה

שקיים חסם שכזה והוא עומק עץ המשחק כולו (כלומר, חישוב עד לעלים של מצבי סיום של המשחק- נצחון או הפסד). לא ציפינו שההפרש בין שני המעברים יהיה גדול.

- MinimaxAgent ו-AlphaBetaAgent קיבלו תוצאות דומות בכל העומקים. זה תואם את הציפייה שלנו שכן אלגוריתם AlphaBeta מחזיר את אותם ערכים שמחזיר אלגוריתם Minimax. ההבדלים הקטנים נובעים מהאלמנטים הרנדומיים במשחק.

אכן התוצאות תואמות את רוב הציפיות שלנו שכן כאשר נבחן לעומק ונבחר את הצעד הטוב ביותר בכל תור, כך נגדיל את סיכויינו לנצח את המשחק וכך נוכל לזכות בניקוד רב יותר.

4. I.



II.

d=1	d=2	d=3	d=4	
0.000107				ReflexAgent
0.00034				BetterAgent
	0.005536	0.03808	0.306268	MinimaxAgent
	0.00293	0.011427427	0.043372	AlphaBetaAgent
	0.005207	0.036051	0.295413	RandomExpectimaxAgent

5. מהגרף והטבלה ניתן להסיק מספר מסקנות:

- ככל שהעומק גדול יותר, כך זמן החישוב הממוצע גדל. זה נכון עבור כל סוכן ותואם את ציפיותינו שכן כאשר אנו מגדילים את העומק של עץ החיפוש אנחנו בעצם בוחנים מספר הולך וגדל בצורה אקספוננציאלית של מצבים. כלומר, זמן החישוב של הערכת המצב הולך וגדל בצורה אקספוננציאלית, בהתאם.

- כאשר לא מסתכלים לעומק, כפי שקורה בסוכנים הפשוטים, הזמן כמעט מיידי. זה בעצם מקרה פרטי של המסקנה הקודמת. במצב שבו לא מפתחים את עץ המצבים אלא מסתכלים רק על הצעד הבא במשחק החישוב הוא יחסית פשטני (הן בהיוריסטיקה המסופקת והן עבור ההיוריסטיקה שהגדרנו) ולכן חישוב ההערכה של הצעד הזה מהיר מאוד, בהתאם לציפיותנו.
- עבור כל העומקים השונים, זמן החישוב של MinimaxAgent גדול מאשר של AlphaBetaAgent. זה כמובן מתאים לציפיות שלנו שכן זוהי בדיוק המטרה של אלגוריתם AlphaBeta – להגיע לתוצאות זהות לשל Minimax אך בזמן מהיר יותר שמושג ע"י גיזום ענפים שלא תורמים לתוצאה הסופית. כלומר, אנחנו מפחים פחות מצבים ולכן זמן החישוב יהיה מהיר יותר.
- זמן החישוב של MinimaxAgent כמעט זהה לזה של RandomExpectimaxAgent בכל עומק. זה תואם את הציפייה שלנו שכן בשני האלגוריתמים המתאימים מפתחים את אותו מספר של צמתים. אמנם שיטת החישוב של הערך המוחזר שונה עבור "תורות" של רוחות – מינימום בminimax וחישוב תוחלת ב-RandomExpectimax אבל כמות הצמתים לפיתוח נשארת זהה בשני המקרים. (זאת לעומת AlphaBeta שגוזם ענפים מיותרים).

6. להלן תוצאות ההרצות המתוארות בשאלה (במשחקים הכוללים 2 רוחות):

Average Avg Turn Time	Average Score	Ghost Agent	Pacman Agent
0.786480585	2160.6	Random Ghost	RandomExpectimaxAgent
0.819705556	1544.0	Directional Ghost	RandomExpectimaxAgent
0.754937786	1741	Random Ghost	DirectionalExpectimaxAgent
0.824316377	1631.6	Directional Ghost	DirectionalExpectimaxAgent

התוצאות שהתקבלו אכן תואמות את הציפיות שלנו. ראשית, נשים לב כי התוצאה הממוצעת הגבוהה ביותר מתקבלת על ידי סוכן RandomExpectimaxAgent שמשחק מול Random Ghost וזה אכן מסביר את הדעת כיוון שישנה התאמה בין איך שפקמן רואה את העולם לבין איך שהסוכנים בעולם מתנהגים בפועל. כלומר, אופן החשיבה של פקמן מתאים לסביבתו. כמו כן, אנחנו מצפים שבמשחק מול רוח Random Ghost התוצאה תהיה גבוהה יותר מאשר מול Directional Ghost אשר מטרתו היא לגרום לפקמן להפסיד.
לאור האבחנה הנ"ל נצפה גם שכלל המשחקים מול Random Ghost יניבו תוצאה גבוהה מאשר מול Directional Ghost ואכן כך הדבר בשני הסוכנים של פקמן.
מבין המשחקים מול Directional Ghost ציפינו לקבל תוצאה גבוהה יותר הסוכן DirectionalExpectimaxAgent שכן כמו במקרה הראשון, סוכן זה משקף נאמנה את המציאות לעומת סוכן RandomExpectimaxAgent שמעריך שרוחות יתקרבו אליו בהסתברות קטנה מזו האמיתית ולכן שם את עצמו בסיכון גבוה יותר.

7. נבחן ראשית את מבנה הלוח. הלוח מאופיין בכך שהוא מאוד קטן וצפוף. כלומר, לפקמן אין הרבה מרחב לבריחה אם רוח כלשהי מגיעה בעקבותיו. בפרט, במשחק עם 2 רוחות ומעלה מרחב התמרון של פקמן קטן מאוד וסביר שייאכל אם לא ידע לצפות במדויק (ולפעמים גם אם כן ידע) לאן הולכות הרוחות. נוסיף גם כי אין כמוסות בלוח ולכן פקמן בסכנה תמידית.

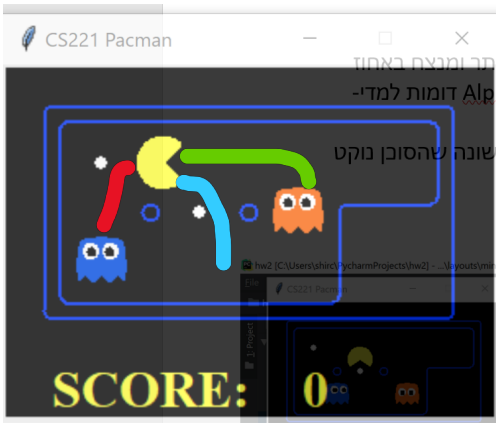
עבור 1000 הרצות של משחק עם לוח minimaxClassic, עומק 4 והגבלה ל-2 רוחות RandomGhost קיבלנו את התוצאות הבאות:

Win Rate	Average Score	
630/1000 (0.63)	141.04	**ReflexAgent**
277/1000 (0.28)	-227.398	**BetterAgent**
824/1000 (0.82)	333.423	**MinimaxAgent**
824/1000 (0.82)	333.495	**AlphaBetaAgent**
878/1000 (0.88)	392.408	**RandomExpectimaxAgent**

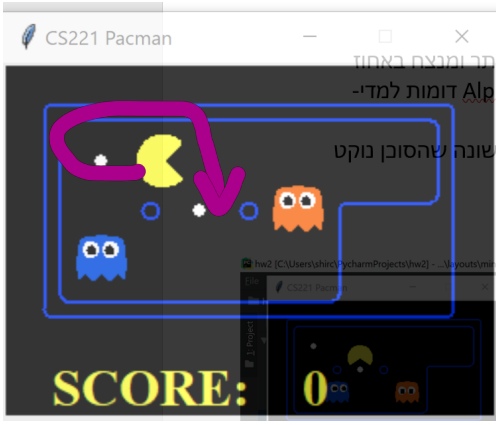
כלומר, נוכחנו לדעת כי RandomExpectimaxAgent הוא הסוכן שמקבל את התוצאה הממוצעת הגבוהה יותר ומנצח באחוז הגדול ביותר של משחקים. נשים לב כי התוצאות של MinimaxAgent ושל AlphaBetaAgent דומות למדי- בדיוק כמו שציפינו בסעיפים קודמים.

בהרצות אלו ניתן לראות כי ReflexAgent מקבל תוצאות גבוהות משל BetterAgent. הסיבה לכך היא ההיוריסטיקה שהגדרנו ל-BetterAgent אשר גורמת לפקמן להפסדים כיוון שבצעד הראשון פקמן פונה ימינה בגלל פרמטר הניידות שהגדרנו וכן בגלל שהרצון לברוח מרוחות גדול בהרבה מהרצון לאכול. הפנייה ימינה והרוחות שמאיימות על פקמן גורמות לו "להתבצר" בצד ימין ולהיתקע שם עד שהוא נאכל על ידי רוח. ReflexAgent מקבל תוצאות נמוכות מיתר הסוכנים כיוון שאינו מסתכל מספר צעדים קדימה.

נשים לב לדפוס באופן הפעולה של MinimaxAgent. בתחילת כל משחק הפעולה הראשונה שהסוכן נוקט היא לפנות ימינה. הסיבה לכך היא שפקמן "מסתכל" 4 צעדים קדימה ולכן, כפי שהוא רואה את העולם- יש לו 2 מסלולים שבהם במקרה הגרוע ביותר (כפי שסוכן minimax מחשב) הוא ימות. מסלולים אלו מסומנים בירוק ואדום בתמונה הבאה. הפעולה הטובה ביותר שהוא יכול לבצע הוא לפנות ימינה ואז לברוח למטה (מסלול כחול.)



מנגד, נשים לב כי מבחינת הלוח הדרך המהירה ביותר לנצחון היא פניה אחת שמאלה ואז חזרה לאוכל האמצעי (מסלול סגול)



לעומת MinimaxAgent, הפעולה הראשונה שסוכן RandomExpectimaxAgent נוקט היא לפנות שמאלה. הסיבה לכך היא שסוכן RandomExpectimaxAgent לא בוחן את המקרה הגרוע ביותר אלא את המקרה הממוצע. במקרה זה, בסיכוי 0.5 הרוח הכחולה תפנה ימינה ולא למעלה ובאותו הסיכוי הרוח הכתומה תפנה למטה ולא למעלה. כאשר מכניסים את השיקול הזה לחישוב של פקמן, פקמן רואה את ההזדמנות לאכול את האוכל שלשמאלו (שמקבל משקל רב בהיוריסטיקה שהגדרנו).
כלומר, RandomExpectimaxAgent פועל באופן דומה לדרך המהירה לניצחון ואילו MinimaxAgent שפונה ימינה מאריך את הדרך. כאשר הדרך המהירה היא כל כך קצרה והלוח כל כך צפוף הפנייה הראשונה הלא נכונה מקטינה משמעותית את הסיכוי לנצחון במשחק.

8. נבחן ראשית את מבנה הלוח. הלוח מאופיין בכך שלמעשה הוא בנוי כשרוך שבו פקמן כלוא בין 2 רוחות. אחת מהרוחות מאוד קרובה אל פקמן בנקודת ההתחלה, השנייה רחוקה יותר ובצידה נמצא האוכל הדרוש לנצחון במשחק. כמו כן, אין כמוסות בלוח ולכן פקמן בסכנה תמידית.

עבור 1000 הרצות של משחק עם לוח trappedClassic, עומק 4 והגבלה ל-2 רוחות RandomGhost קיבלנו את התוצאות הבאות:

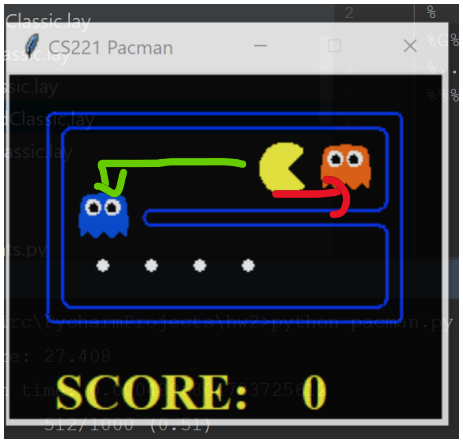
Win Rate	Average Score	
26/1000 (0.03)	-474.713	**ReflexAgent**
491/1000 (0.49)	5.694	**BetterAgent**
0/1000 (0.00)	-501.0	**MinimaxAgent**
0/1000 (0.00)	-501.0	**AlphaBetaAgent**
507/1000 (0.51)	22.238	**RandomExpectimaxAgent**

כלומר, נוכחנו לדעת כי RandomExpectimaxAgent מקבל את התוצאה הממוצעת הגבוהה ביותר ומנצח באחוז הגדול ביותר של משחקים. למעשה, MinimaxAgent מפסיד בכל 1000 המשחקים ומפסיד מיד עם תחילת המשחק. נשים לב כי התוצאות של MinimaxAgent ושל AlphaBetaAgent זהות- בדיוק כמו שציפינו בסעיפים קודמים.

במקרה זה נשים לב כי ReflexAgent ו-BetterAgent מקבלים תוצאות גבוהות משל MinimaxAgent ושל AlphaBetaAgent. במקרה זה ההסתכלות לעומק של MinimaxAgent היא זו שפוגעת בו ודווקא ReflexAgent ו-BetterAgent שאינם חושבים מה יקרה בעוד מספר צעדים זוכים מכך שכן הם אינם רואים את המקרה הגרוע ביותר – מוות של פקמן. התוצאה של BetterAgent גבוהה משל ReflexAgent עקב בחירת ההיוריסטיקה

שלנו. BetterAgent יודע לברוח מהרוח הימנית שקרובה אליו ולא מתייחס לרוח השניה ובנוסף הוא מנסה להתקרב לאוכל ולכן פונה שמאלה מיד עם תחילת המשחק. לעומתו, ReflexAgent לא מנסה להגיע לאוכל ולכן מת בהסתברות גבוהה יותר.

נשים לב לדפוס באופן הפעולה של MinimaxAgent. בתחילת כל משחק הפעולה הראשונה שהסוכן נוקט היא לפנות ימינה. הסיבה לכך היא שפקמן "מסתכל" 4 צעדים קדימה ולכן רואה כי יש לו 2 מסלולים אפשריים שבשניהם במקרה הגרוע ביותר (כפי שסוכן minimax מחשב) הוא ימות. מסלולים אלו מסומנים בירוק ואדום בתמונה הבאה. במקרה זה, במסלול האדום פקמן ימות תוך צעד אחד ולכן "ייקנס" בנקודה לפני שימות בעוד שבמסלול הירוק פקמן ימות תוך 3 צעדים ולכן "ייקנס" ב-3 נקודות. כלומר, פקמן הולך למות בכל מקרה ולכן מעדיף למות כאשר יש לו ניקוד גבוה יותר ופונה ימינה.



לעומת MinimaxAgent, הפעולה הראשונה שסוכן RandomExpectimaxAgent נוקט היא לפנות שמאלה. הסיבה לכך היא שסוכן RandomExpectimaxAgent לא בוחן את המקרה הגרוע ביותר אלא את המקרה הממוצע. במקרה זה, ייתכן מצב שבו פקמן לא ימות כלל, אם הוא יפנה שמאלה והרוח הכחולה תחליט לפנות למטה.
כלומר, במצב ההתחלה בסיכוי 0.5 הרוח הכחולה תפנה שמאלה ולא למעלה. ולכן כעת פקמן צריך לבחור האם "להתאבד" ע"י פנייה ימינה או לפנות שמאלה ולקוות שהרוח הכחולה לא תבוא לכיוונו. אכן, בתוחלת ב50% מהפעמים הרוח תבחר ללכת בכיוון המנוגד לפקמן וכך הוא יוכל לאכול את כל האוכל ולנצח. זה תואם את התוצאות שקיבלנו שבהן ב51% מהמשחקים סוכן RandomExpectimaxAgent ניצח.

9. ראשית כל, כיוון שכל הסוכנים מלבד $ReflexAgent$ משתמשים בהיוריסטיקה החדשה שהגדרנו בחלק ב' נרצה להשוות בין היוריסטיקה זו לבין זו שסופקה עם התרגיל. לשם כך, נרצה להשוות בין ביצועי $ReflexAgent$ בלוחות השונים לעומת ביצועי $BetterAgent$ כיוון שמלבד אופן חישוב ההיוריסטיקה אופן הפעולה שלהם זהה. נבחן את התוצאות שקיבלנו ונמצאות בקובץ experiments.csv:

	ReflexAgent	BetterAgent
capsuleClassic	-487.286	86.42857
contestClassic	239.4286	838.4286
mediumClassic	-121.143	1684
minimaxClassic	83.14286	74.42857
openClassic	242.5714	1390.857
originalClassic	-283.857	909.1429
smallClassic	-284.857	1150

testClassic	519.1429	524.8571
trappedClassic	-501.714	88.85714
trickyClassic	-36.7143	186.8571

ניתן לראות כי מלבד הלוח minimaxClassic, בכל המשחקים על יתר הלוחות (שבוצעו תחת התנאים הנדרשים בסעיף 1) אשר משתמשים ב-$BetterAgent$ מתקבלת בממוצע תוצאה גבוהה יותר, ולרוב משמעותית, מאשר זו המתקבלת בשימוש ב-$ReflexAgent$.
עבור הלוח minimaxClassic נשים לב כי אכן ההיוריסטיקה שלנו גורמת לפקמן להפסדים כיוון שבצעד הראשון פקמן פונה ימינה בגלל פרמטר הניידות שהגדרנו וכן הגלל שהרצון לברוח מרוחות גדול מהרצון לאכול. תמונה של הלוח מופיעה בסעיף 7 בפרק זה ויחד איתה ההסבר מדוע הפנייה ימינה גוררת הפסד בהסצברות גבוהה. סך הכל נסיק כי ההיוריסטיקה שלנו מתגברת על ההיוריסטיקה הפשוטה ונמליץ להשתמש בה כחלופה להיוריסטיקה הפשוטה.
נבחין עוד כי חישוב ההיוריסטיקה החדשה אורך זמן רב יותר מהפשוטה אך ההפרש בין ממוצע הזמן הממוצע לתור הארוך ביותר של ההיוריסטיקה שלנו לממוצע הזמן הממוצע הקצר ביותר של ההיוריסטיקה הפשוטה הינו 0.000681354 שניות ולכן בשימוש באלגוריתמיים שמסתכלים לעומק, עבור עומקים שאינם גדולים מדי (עד עומק 3) נוכל "לספוג" את זמן החישוב.

נבצע כעת ניתוח מעמיק בין זוגות השחקנים, כאשר נשווה שני פרמטרים בין כל זוג שחקנים.
1. תוצאה ממוצעת על כלל הלוחות.
2. זמן ממוצע ממוצע לתור.

BetterAgent Vs. ReflexAgent:
הניתוח ששימש לבחינת ההיוריסטיקה הוא בדיוק הניתוח שמשווה בין שני הסוכנים הללו ומופיע לעיל.

לאור האמור לעיל, כיוון שהראנו שההיוריסטיקה שלנו עדיפה על זו המסופקת בתרגיל (למעט לוחות מאוד קטנים שבו פקטור הניידות נכנס כבעל משקל גבוה, בעוד שברוב הלוחות היינו רוצים שייכנס עם משקל נמוך ביחס ליתר הפרמטרים), ושני הסוכנים ReflexAgent ו-BetterAgent פועלים על פי אותו אלגוריתם חיפוש נשווה מעתה ואילך רק את הסוכן BetterAgent ליתר הסוכנים. כפי שנראה, יתר הסוכנים עדיפים כולם מבחינת תוצאות המשחק על פני BetterAgent ולפיכך גם על פני ReflexAgent. באופן דומה, הזמן הממוצע הממוצע לתור של כל יתר הסוכנים יהיה ארוך משל BetterAgent וכיוון שזמן החישוב ההיוריסטיקה החדשה גדול מזו שסופקה בתרגיל נוכל לקבוע כי ReflexAgent הוא הסוכן המהיר ביותר מבין כלל הסוכנים.

BetterAgent Vs. MinimaxAgent:
1. תוצאה ממוצעת – ניתן לראות בקובץ experiments.csv שהתוצאות של סוכן ה-Minimax טובות משמעותית מהתוצאות של BetterAgent. ניתן להסביר זאת עקב העובדה שסוכן ה-Minimax מבצע הסתכלות לעומקים שונים: 2,3,4 בעוד שסוכן BetterAgent אינו מבצע הסתכלות זו ומתייחס למהלך הבא בלבד.
2. זמן ממוצע ממוצע לתור – במקרה הנ"ל הזמן הממוצע לתור של MinimaxAgent פחות טוב משמעותית אל מול BetterAgent. הסיבה לכך היא אותה הסיבה שגרמה ליחס הפוך בין הסוכנים בתוצאות המשחק. אלגוריתם המינימקס שמחשב מספר צעדים קדימה נדרש לפתח כמות הולכת וגדלה אקספוננציאלית של צמתים עבור כל רמת עומק נוספת. כמות ההפעלות של ההיוריסטיקה באלגוריתם זה היא כמספר העלים בעץ החיפוש ועל כן, בעוד שסוכן BetterAgent מחשב את ההיוריסטיקה פעם אחת לכל מהלך אפשרי, סוכן MinimaxAgent מחשב אותה בסדר גודל גדול בהרבה (שתלוי בנוסף גם במספר הרוחות ומספר הצעדים האפשריים של כל אחת מהן).

BetterAgent Vs. AlphaBetaAgent:

1. תוצאה ממוצעת – כפי שנראה בהמשך, ובהתאם לנכונות אלגוריתם אלפא-בטא, התוצאה הממוצעת של AlphaBetaAgent מתכנסת לאותה אותה כמו של MinimaxAgent. לכן כמו בהשוואה האחרונה התוצאה הממוצעת של סוכן AlphaBetaAgent גבוהה משל BetterAgent, וזאת בזכות הבחירה המיודעת יותר של מהלכים שנובעת מהסתכלות של מספר צעדים קדימה.
2. זמן ממוצע ממוצע לתור – במקרה הנ"ל הזמן הממוצע הממוצע לתור של AlphaBetaAgent פחות טוב (על כלל העומקים) אל מול BetterAgent שכן חישוב פלט האלגוריתם לוקח יותר זמן ב-AlphaBetaAgent והסתכלות לעומק מוסיפה זמן רב.

BetterAgent Vs. RandomExpectimaxAgent:

בדומה להשוואה מול הסוכנים MinimaxAgent ו-AlphaBetaAgent התוצאה הממוצעת למשחק של RandomExpectimaxAgent גבוהה משל BetterAgent והזמן הממוצע הממוצע לתור ארוך יותר גם כן. הסיבות גם הן אותן סיבות, הנובעות מהסתכלות לעומק של האלגוריתם Expectimax.

MinimaxAgent Vs. AlphaBetaAgent:

1. תוצאה ממוצעת – ניתן לראות בגרף מסעיף ב' כי התוצאה הממוצעת עבור שני הסוכנים הנ"ל הינה דומה מאוד. הסיבה לכך שהתוצאות אינן זהות היא שבמהלך המשחק פקמן נדרש לעיתים לבחור באופן אקראי מבין מספר מהלכים בעלי ערך זהה וכיוון שההרצות לא בוצעו עם ערך SEED קבוע ייתכן כי פונקציית random הניבה פלטים שונים. עם זאת, נוכל לראות כי בסעיפים 7,8 ערכנו בדיקות נוספות והרצנו 1000 משחקים עם תנאים זהים מלבד הסוכן של פקמן וקיבלנו כי הערכים של AlphaBetaAgent ו-MinimaxAgent מתכנסים לאותו הערך של ממוצע תוצאות ולאותו אחוז הצלחה במשחקים. אף על פי שבסעיף 8 הבדיקה דיי מנוונת כי שני הסוכנים מתים מיד, בסעיף 7 הסוכנים אכן מצליחים להגיע לתוצאות מספריות שאינן טריוויאליות. כמובן שהסיבה להתכנסות התוצאות של שני הסוכנים לאותה תוצאה ממוצעת היא נכונות אלגוריתם אלפא-בטא וה"הבטחה" שלו להחזיר את אותם ערכים שאלגוריתם מינימקס היה מחזיר.
2. זמן ממוצע ממוצע לתור – כפי שניתן לראות בגרף מסעיף ד' סוכן AlphaBetaAgent פועל בזמן טוב משמעותית מאשר סוכן ה-MinimaxAgent . הסיבה לכך היא כמובן פעולת הגיזום של אלגוריתם אלפא-בטא שמאפשרת לו להימנע מחישובים אשר אינם משפיעים על הפלט הסופי של האלגוריתם.

לכן ניתן לקבוע שמבין שני סוכנים אלו, העדיפות הברורה היא לסוכן AlphaBetaAgent .

MinimaxAgent Vs. RandomExpectimaxAgent:

1. תוצאה ממוצעת – בהשוואה זו נקבל כי עבור עומק 2 סוכן RandomExpectimaxAgent מניב תוצאה גבוהה יותר יחסית באופן משמעותי ועבור עומקים 3,4 ההפרש בין הסוכנים קטן (בעומק 3 ההבדל לא ניכר כלל). כלומר, בהסתכלות כוללת נקבל כי RandomExpectimaxAgent אכן עדיף על MinimaxAgent בהקשר זה.
2. זמן ממוצע ממוצע לתור – זמן החישוב של MinimaxAgent כמעט זהה לזה של RandomExpectimaxAgent בכל עומק. הסיבה לכך היא ששני האלגוריתמים המתאימים מפתחים את אותו מספר של צמתים. אמנם שיטת החישוב של הערך המוחזר שונה עבור "תורות" של רוחות – מינימום בminimax וחישוב תוחלת ב-RandomExpectimax אבל כמות הצמתים לפיתוח נשארת זהה בשני המקרים.

AlphaBetaAgent Vs. RandomExpectimaxAgent:
בדומה להשוואה הקודמת, ועל סמך ההסברים שסופקו בהשוואה שבין הסוכנים MinimaxAgent ו-AlphaBetaAgent. התוצאה הממוצעת למשחק של RandomExpectimaxAgent גבוהה (אך לא בהפרש ניכר) משל AlphaBetaAgent והזמן הממוצע הממוצע של AlphaBetaAgent קטן משמעותית משל RandomExpectimaxAgent. במקרה זה, אם המשחק לא מוגבל בזמן חישוב נעדיף להשתמש בסוכן RandomExpectimaxAgent על מנת לקבל תוצאות טובות יותר. אחרת, נעדיף לשחק עם AlphaBetaAgent אשר ביצועיו לא פחותים בהרבה אך בזמן החישוב הוא משמעותית מהיר יותר.

RandomExpectimaxAgent Vs. <u>Directional</u>ExpectimaxAgent:
מקרה זה הינו יותר מורכב מקודמיו שכן לעומת ההשוואות הקודמות, שבוצעו כולן על סמך משחקים מול רוחות רנדומליות, ההשוואה הזאת נעשית אינה הוגנת ברגע שמודדים את הסוכן מול רוח שאינה מתאימה לו (סוכן RandomExpectimaxAgent מול רוחות רנדומליות, ולהפך). ההשוואה פורטה בהרחבה בסעיף 6 אך נחזור כאן על עיקרי הדברים.

1. תוצאה ממוצעת – קיבלנו כי עבור רוח שמתאימה לסוכן שלה RandomExpectimaxAgent מקבל תוצאה גבוהה יותר מאשר DirectionalExpectimaxAgent והסיבה לכך היא שבמשחק מול רוחות רנדומליות הסיכוי שלהן לתקוף את פקמן קטן וכך סיכויי ההשרדות שלו עולים ויחד איתם הסיכוי לנצח במשחק. (באופן, דומה עולה הסיכוי שלו לאכול אותן כשהן מפחדות.)
2. זמן ממוצע ממוצע לתור – במקרה הנ"ל הזמן הממוצע הממוצע לתור של שני הסוכנים הינו דומה למדי. אמנם חישוב של DirectionalExpectimaxAgent דורש חישוב על התפלגות התנועה של כל רוח אבל חישוב זה אינו דורש פעולות "יקרות" וראינו כי ההבדל לא ניכר בתוצאות הזמנים.

לסיכום, אם אנחנו יודעים מול איזה סוג רוח אנו משחקים, נמליץ להשתמש בסוכן המתאים לה. אחרת, נמליץ להשתמש בסוכן DirectionalExpectimaxAgent שכן הוא ישים את פקמן "בצד הבטוח" של המשחק וראינו, לאור תוצאות סעיף 6, כי הממוצע שלו עבור 2 סוגי הרוחות גבוה מאשר של RandomExpectimaxAgent.
מכיוון שהניסויים על סוכנים קודמים נעשו מול רוחות רנדומליות בלבד, ועבורן הסוכן RandomExpectimaxAgent הוכיח עצמו כאופטימלי, לא נשווה בין יתר הסוכנים לסוכן DirectionalExpectimaxAgent.

כעת נבצע ניתוח פשטני של לוחות המשחק השונים. ניתוח מורחב של שניים מהלוחות מופיע בסעיפים קודמים.

capsuleClassic :
השחקן הכי טוב ללוח זה הינו: AlphaBetaAgent בעל עומק 4.
לוח זה מתאפיין במספר מועט של קירות ומספר רב של "נתיבי בריחה" עבור פקמן בעת הרצתו של המשחק. לפיכך, יהיה לפקמן מספר רב של אפשרויות לבחירת מהלך בכל תור ולכן כאשר העומק יהיה 4 נקבל את התוצאות הגבוהות ביותר. ראוי לציין כי בתחילת המשחק הרוחות רחוקות מפקמן לכן הוא יכול להיערך בהתאם ולבחון את צעדיו כבר מההתחלה.

contestClassic:
השחקן הכי טוב ללוח זה הינו:RandomExpectimaxAgent בעומק 4.
הלוח מאופיין בגודלו הגדול. גודלו הגדול נותן יתרון לאלגוריתמים המסתכלים לעומק שכן אלו יבחנו את צעדיהם ויקחו את הטוב ביותר. מכיוון שהרוחות אשר שיחקנו מולן הם randomGhost אזי הסוכן ששיחק הכי טוב הוא randomExpectimax שכן הוא מותאם לסביבתו בצורה המיטבית.

mediumClassic:
השחקן הכי טוב ללוח זה הינו:RandomExpectimaxAgent בעומק 4.
ניתוח הלוח הנ"ל יהיה דומה ללוח הקודם (contestClassic). שני לוחות אלו יחסית דומים, הן בגודלים והן במבניהם. התוצאות שומרות על אותו היחס בין כלל הסוכנים כאשר הן בהיסט של 200 נקודות פחות בממוצע לכל סוכן.

minimaxClassic:
אמנם בקובץ ה-excpermints.csv הסוכנים הטובים ביותר ללוח זה הם minimaxAgent ו-alphabetaAgent בעלי עומק 4 אך בסעיף 7 בוצע ניתוח מעמיק ללוח זה שהצביע על העובדה כי דווקא RandomExpectimaxAgent מרוויח תוצאות טובות יותר לאורך זמן. מאפייני הלוח תוארו גם הם בהרחבה בסעיף 7.

openClassic:
השחקן הכי טוב ללוח זה הינו RandomExpectimaxAgent בעומק 3.
עם זאת, התוצאות של כלל הסוכנים בלוח זה דומות מאוד, גם של זו שאינה מסתכלת לעומק אך משתמשת בהיוריסטיקה שלנו. הסיבה לכך היא שמבנה הלוח פשוט מאוד (ללא קירות) ויש רק רוח אחת במשחק. לכן, ההיוריסטיקה שלנו יודעת שעליה להיזהר מהרוח היחידה ולהשיג אוכל בעוד פרמטרים אחרים אינם משמעותיים (אין אלמנט של ניידות, יש כמוסה אחת בלבד).

originalClassic:
השחקן הכי טוב ללוח זה הינו AlphaBetaAgent בעומק 3.
הלוח מאופיין במספר רב של כמוסות וגודלו גדול מאוד. עקב מיקומן ההתחלתי של הרוחות לפקמן יש יכולות מילוט רבות ולכן הוא מגיע לתוצאות גבוהות. התוצאות של כל הסוכנים שמסתכלים לעומק דומות וגבוהות משמעותית מאלו של הסוכנים שאינם מסתכלים קדימה.

smallClassic:
השחקן הכי טוב ללוח זה הינו AlphaBetaAgent בעומק 3.
הלוח מאופיין במספר זהה של כמוסות אל מול רוחות. גודלו יחסית מרווח, לכן לפקמן יש יכולות תמרון לאכול ולא להיאכל. סוכנים אשר מסתכלים לעומק כמובן יצליחו בלוח זה יותר מאשר סוכנים שאינם מסתכלים, אך מכיוון שאין הרבה קירות וגודלו של הלוח גדול, אזי גם הסוכן הפשוט אשר משתמש בהיוריסטיקה החדשה שלנו משיג תוצאות דומות לאלו של הסוכנים שמסתכלים לעומק. הסיבה לכך היא שלפקמן יש יכולות מילוט רבות בלוח זה ומספר כמוסות רב.

testClassic:
עבור לוח זה קיבלנו שכל הסוכנים קיבלו תוצאות מאוד דומות עבור כלל העומקים. הסיבה לכך היא כנראה מימדי הלוח הקטנים. כלל הסוכנים פועלים יחסית באותה הצורה; ראשית לא להיאכל על ידי הרוח הבודדה אשר יש בלוח, ולאחר מכן לאכול את האוכל שיש. בלוח זה אין כמוסה, אזי אין דרך לאכול את הרוח היחידה שיש.

trappedClassic:
השחקן הכי טוב ללוח זה הינו: RandomExpectimaxAgent. ביצוע מעמיק בוצע בסעיף 8.

trickyClassic:
השחקן הכי טוב ללוח זה הינו AlphaBetaAgent בעומק 3.
הלוח מאופיין בכך שהוא יחסית גדול ומכיל הרבה כמוסות. מכיוון שהלוח גדול ומכיל כמוסות רבות, סוכנים אשר מסתכלים לעומק ירוויחו במקרה זה.

סיכום - השפעת הלוחות:
ככלל, ניתן לראות שככל שהלוח גדול יותר, כך ניקוד הסוכנים אשר מסתכלים לעומק גבוה יותר ביחס לאלו שאינם מסתכלים. כמות הכמוסות והיחס שלהן אל מול מספר הרוחות גם הוא משפיע על הניקוד הסוכנים. ככל שיש יותר כמוסות על הלוח (ולפחות רוח אחת) כך גדל הסיכוי של פקמן לאכול את הרוח ולכן התוצאה הסופית תהיה גדולה בממוצע יותר עבור מספר כמוסות גדול יותר. בנוסף, ככל שישנן יותר רוחות נרצה שהיחס בין כמות הכמוסות לרוחות יגדל גם כן. כך למשל בלוח עם 2 רוחות, נעדיף 2 כמוסות אל מול כמוסה יחידה שכן פיזור הרוחות על הלוח עלול לגרום לכך שכמוסה יחידה לא תספיק לאכילת שתי הרוחות.
לכן, כפי שניתן להבין, גודל הלוח ומספר הכמוסות בו משפיעים בצורה ישירה על ציון הסוכנים, כאשר ככל שהסוכן מעמיק יותר כך ציונו יהיה גדול יותר, אם שני פרמטרים אלו יהיו גדולים יותר.
בנוסף, גם למספר הרוחות כמובן יש השלכה על התוצאה הסופית והיא תלויה במספר הכמוסות. אם אין כמוסות כלל- ברור שנוכחות של רוחות תכניס את פקמן לסכנה ותקטין את תוצאתו הממוצעת. מנגד, אם ישנן כמוסות ופקמן יצליח לאכול את הרוחות הניקוד שלו במשחק יעלה בצורה משמעותית.

נתאר כעת תרחיש שבו לא הייתה קיימת מגבלת עומק על הסוכנים, אלא מגבלה בזמן החישוב. שינוי המגבלה זו היה מציב לנו tradeoff שבין איכות ההיוריסטיקה לבין עומק החיפוש שהיינו יכולים להגיע אליו. במצב של מגלת זמן, היינו משתמשים באלגוריתם חיפוש לעומק מסוג anytime אשר היה מפתח לעומק מסוים, שומר את התוצאה הטובה ביותר שמצא וממשיך לחפש בעומקים הולכים וגדלים. במצב זה, ייתכן והיינו מעדיפים להחליף את ההיוריסטיקה של הערכת המצבים להיוריסטיקה פשוטה יותר שכן זו הייתה מאפשרת לאלגוריתם anytime להגיע לעומקים גדולים. מנגד, היוריסטיקה מורכבת יותר, שזמן חישובה ארוך יותר, הייתה מגבילה את העומק בו היינו יכולים לחפש במגבלת הזמנים אך התוצאה שאליה היינו מגיעים בעומקים שכן חישבנו הייתה ככל הנראה מדוייקת יותר. לכן, נוכל לשער כי במשחקים שמשתמשים בלוחות קטנים כיוון שההסתכלות לעומק מאבדת ממשמעותה החל מעומק מסויים נעדיף להשתמש בהיוריסטיקה מורכבת בעוד בלוחות גדולים נעדיף להשתמש בהיוריסטיקה פשוטה יותר שמחשבת הרבה צעדים קדימה.

ניתן להכליל את טיעון זה ולטעון שככל שהלוח גדול יותר, הסתכלות לעומק גדולה יותר מניבה תוצאה סופית גבוהה יותר.

מספר מסקנות כוללות נוספות:
1. אין כל סיבה הגיונית, למעט אם נתבקשנו לכך, להשתמש בסוכן minimax. כפי שהוכח בהרצאה ובתרגול, וכן הוכח מעשית בתרגיל בית זה, ערכי ההחזרה של שני סוכנים אלו הם זהים, אך זמן הריצה של סוכן alpha-beta נמוך משמעותית מסוכן הminimax לכן אין כל סיבה להשתמש בו. אמנם בבדיקת הלוחות ראינו יותר מקרים שבהם alpha-beta קיבל תוצאות טובות יותר אבל הבדיקות כללו 7 משחקים בלבד על כל סט של תנאים ולכן אינם מהווים מדגם מייצג טוב מספיק. בריצות נוספות שהראינו לאורך התרגיל ראינו כי שני הסוכנים מתכנסים לאותו ערך בדיוק.
2. כאשר אנו יודעים מראש באיזו סביבה פקמן הולך לשחק (מבחינת מבנה הלוח, מול אילו רוחות וכו') מומלץ להתאים את הסוכן שיפעל בסביבה אשר תואמות לו, כלומר אם הרוחות פועלות במצב של directional לדוגמא, אזי מומלץ שהסוכן גם הוא יפעל במצב שבו הוא מותאם אליהן שכן שיטה זו הוכיחה את עצמה כמועילה ביותר.
התאמת הסוכן לסביבה תניב תוצאות טובות יותר מאשר סוכן שאינו מותאם לסביבתו.
3. ישנו חסם אסימפטוטי על עומק החיפוש שנמצא כמועיל עבור סוכן בלוח כלשהו. ככל שנעמיק יותר נצפה (לרוב) שהתוצאות ישתפרו אם כי ישנו חסם שבו נוכל להגיע לעלים בעץ החיפוש (מצב של הפסד או נצחון) ולכן הגדלת עומק החיפוש מעבר לסף זה יהיה מיותר. בנוסף, נבחין כי ישנם לוחות, כדוגמת trappedClassic, שבהם החיפוש לעומק

הוא דווקא זה שגורם לסוכנים כמו Minimax "להתאבד" על הצעד הראשון ולו היינו בוחנים רק צעד אחד קדימה היינו מגדילים את תוצאת המשחק הממוצעת.
עוד בהקשר זה- נזכור כי כלל האלגוריתמים מומשו על סמך ההנחה שהוצגה בחלק ג' ולפיה המשחק מנוהל בתורות. כיוון שהנחה זו אינה תואמת את המציאות נוכל לנחש גם שהיא תורמת מידה כלשהי של "רעש" לחישוב הערכת הצעדים ושלעומקים גדולים מאוד "רעש" זה יגדל ויפגום בתוצאות. כיוון שבניסויים שלנו הגענו לעומק מקסימלי של 4 והגדלת העומק לעומקים גדולים מאוד תהפוך את זמן הריצה ללא סביר לא נבדוק את הטענה ונשאיר אותה כאן כהשערה.

4. לגודל הלוח ישנה חשיבות מכרעת אל מול התוצאות עבור הסוכנים. ככלל, סוכנים אשר מסתכלים לעומק יטיבו יותר עם לוח גדול מאשר לוח קטן. לוח קטן לוקח בצורה יחסית את היתרונות של הסתכלות לעומק, ומכניס יותר מימד של אקראיות לאופן המשחק. לכן ניתן להכליל טענה זו ולטעון שככל שמרחב הפעולה של הסוכן שלנו גדול יותר, כך סוכן אשר מסתכל לעומק יהיה בעל תוצאות טובות יותר, מאשר סוכן שאינו מסתכל לעומק. בלוחות קטנים יותר, פערים אלו מצטמצמים משמעותית עד כמעט שאינם קיימים כפי שנוכחנו לדעת בלוח minimaxClassic עבור 1000 הרצות, קיבלנו תוצאות מאוד דומות.
5. היוריסטיקה אשר תוכיח את עצמה כמוצלחת בסוכנים שאינם מסתכלים לעומק תנבא הצלחה גם עבור סוכנים אשר משתמשים באלגוריתמים מורכבים יותר אשר משתמשים בעץ חיפוש.

<u>חלק ט'</u>

בחרנו שסוכן התחרות שלנו ישתמש לרוב בווריאציה של אלגוריתם אלפא בטא ובמקרים מסויימים בווריאציה של RandomExpectimax. ראשית, נעדיף לרוב להשתמש בסוכן המשתמש באלגוריתם אלפא בטא לעומת סוכן המשתמש באלגוריתם minimax כיוון שאלו צפויים לקבל את אותן התוצאות אך בהפרש זמן ניכר. כיוון שזמן המשחק כולו מוגבל ל-30 שניות, כמובן שנעדיף את האלגוריתם המהיר מבין השניים. שנית, אמנם ראינו כי בממוצע התוצאות ישנה עדיפות דווקא לסוכן Expectimax אך זמן החישוב שלו דומה לשל minimax ואורך זמן רב. לכן, הסקנו שאיכות הפתרון בשקלול של זמן החישוב והתוצאה הממוצעת היא אופטימלית בשימוש באלפא-בטא.

עקב מגבלות הזמן של התרגיל הגבלנו את עומק החישוב של הסוכן שלנו לעומק 2 בלבד עבור סוכן אלפא-בטא ועומק 3 עבור סוכן RandomExpectimax המשחק בלוחות קטנים בלבד. לשם חישוב הערכת המצבים בעלים של עץ החיפוש החלטנו להשתמש בהיוריסטיקה אשר הגדרנו בחלק ב' ללא פרמטר הניידות אשר בדיעבד לא תרם לתוצאות המשחקים ויצר מצבי קיצון בעייתיים. בנוסף, בכדי לשפר את תוצאות המשחק ערכנו שינוי קל באלגוריתם שמומש בסעיפים קודמים. במהלך אלגוריתם אלפא-בטא בכל אחת מהשכבות צריך האלגוריתם לחשב את המהלכים הבאים האפשריים של הסוכן. כעת, טרם המעבר על המהלכים אנחנו "מערבבים" את רשימת המהלכים. בחינת תוצאות האלגוריתם ברור כי נכונותו לא השתנתה שכן הערכים הינם אותם ערכים רק שעוברים עליהם בסדר שונה. מנגד, אם ישנם מספר צמתים בעלי אותו ערך כעת ייתכן ונגזום צמתים שונים בכל פעם- כתלות בסדר המעבר עליהם. אנחנו צופים שפעולה זו תמנע מצב שבו פקמן נתקע על סדרת צעדים קבועה.

שמנו לב כי החזרה על סדרת צעדים קבועה מתרחשת לעיתים ומעכבת את התקדמות המשחק, מעבר להגבלת הזמן שהוצגה. לשם כך, הוספנו אלמנט נוסף של אקראיות לסוכן שלנו. האלגוריתם שלנו פועל כך שבתחילת כל פעולה הוא מגריל מספר בטווח 1-45. אם התקבל הערך 45 האלגוריתם לא מפעיל את אלפא בטא אלא מגריל צעד התקדמות מבין סט הצעדים החוקיים. אחרת, מפעיל את אלגוריתם אלפא בטא ומחזיר את התוצאה שהתקבלה. בצורה זו אנו מבטיחים שבממוצע- כל 45 פעולות של פקמן יבוצע מהלך אקראי אחד. אנחנו מצפים שהמהלך האקראי של פקמן יאפשר לו "להשתחרר" מסדרת הפעולות הקבועה שנקלע אליה. באמצעות שיטה זו אנו מוסיפים מימד של אקראיות אשר מניב לנו בממוצע זמן ריצה טוב יותר תוך פגיעה מינימלית באיכות התוצאות. הערך 45 נבחר ע"י ניסוי וטעייה לאחר שהצליח לעמוד בדרישת הזמנים אך לא לפגוע משמעותית בטיב הפתרון.

יחד עם האמור לעיל, עבור לוחות קטנים (=ששטחם לכל היותר 50 יחידות) בחרנו דווקא שכן להשתמש באלגוריתם Expectimax מתוך ההנחה שבלוחות אלו זמן המשחק יהיה מהיר בין כה וכה ולכן נעדיף את התוצאות הטובות יותר. ראינו גם בסעיפים קודמים כי בלוחות אלו אכן סוכן Expectimax היה טוב יותר בצורה ניכרת. את אותו "ערבוב" שיישמנו עבור אלפא בטא יישמנו גם עבור Expectimax אך ויתרנו על התורות האקראיים שכן בלוח קטן גם צעד קטן בכיוון לא נכון יכול לעלות לפקמן בחייו.