חלק א' – הכרות עם המשחק

**תרגיל 3 – Reflex Player**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 First ומעריך מצבים על ידי הניקוד שהם מניבים.

חלק ב' – בניית סוכן משופר

**תרגיל 1 – Better Evaluation Function**  
  
הפונקציה שהגדרנו הינה שילוב של הפרמטרים הבאים:

* הוספת ניקוד המצב.
* הוספת ממוצע הניקוד של המצבים העוקבים למצב.
* עבור כל רוח במרחק מנהטן לכל היותר 3 מפאקמן, אם היא מפוחדת לפחות ל 3 צעדים הבאים נוסיף את המרחק להערכת המצב אחרת נחסר.
* בניית מסלול הקצר ביותר לאסיפת כל האוכל הנותר בלוח המבוסס מרחקי מנהטן. וחיסור אורך המסלול מערך המצב (ככל שיותר קטן יותר טוב).
* חיסור של מרחק מנהטן מהקפסולה הכי קרובה לפאקמן (ככל שיותר קטן יותר טוב).
* חיסור של 2 נקודות אם הפאקמן נמצא בין שני קירות מאוזנים.
* חיסור של 2 נקודות אם הפאקמן נמצא בין שני קירות מאונכים.

**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 את המצב לפיו. אם הרוח מפוחדת והיא קרובה לפאקמן, מרחק מנהטן 3, אזי נרצה להגדיל את הניקוד עבור מצב זה ונוסיף את המרחק מנהטן זה, אחרת היא קרובה לפאקמן אך לא מפוחדת ולכן מאיימת עליו ורצה לחסר את המרחק מנהטן ממנה.  
על מנת לצבור יותר נקודות הפונקציה מעריכה את המצב גם לפי המסלול לאסיפת כל הנקודות ה**אוכל על הלוח**, ככל שהמסלול קצר יותר כך הפונקציה תחסר פחות מהערך הסופי והוא יגדל.  
בנוסף אנו רוצים לתת לפאקמן יותר נתיבי מילוט מהרוחות ולכן נעדיף שלא ימצא בין שני **קירות** ולכן עבור מצבים כאלו נחסר 2 נקודות.  
המוטיבציה מאחורי היוריסטיקה היא מתן ניקוד גבוה יותר למצבים שרחוקים מרוחות וקרובים לאוכל ולרוחות מפוחדות 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, זזות כל אחת בתורה אמנם במשחק זה כל הסוכנים מבצעים מהלך ביחד ואין מדובר במשחק תורות ובפרט תזוזות הרוחות אינה משפיעה אחת על השנייה.

**תרגיל 3 – מימוש נוסף**במקום לממש שכבה עבור כל סוכן שמשתתף, כלומר עבור כל רוח, ניתן בשכבה אחת לייצג את כל הרוחות. בשכבה זו כל קשת אשר יוצאת מהמצב מגלמת פרמוטציה של מהלכים חוקיים עבור כל הרוחות. בשיטה זו יש לנו 2 סוכנים הפאקמן והרוחות אשר מיוצגות על יד סוכן יחיד.  
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 בעומק 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 דבר שלא יעיל לדוגמא עבור האלגוריתם Alpha - Beta אשר מספר הפיתוחים מושפע ממקדם הסיעוף.

חלק ד' – בניית סוכן Alpha – Beta  
**תרגיל 1 – מבנה העץ**מבנה העץ שהוגדר בתרגיל 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, במבנה אחר לדוגמא כמו ההצעה בסעיף קודם יש לכאורה 2 סוכנים, הפאקמן ושאר הרוחות לכן במקרה הממוצע יבצע  כלומר יותר פיתוחים.  
מסקנה: יש השפעה על מבנה העץ שהוגדר על כמות הפיתוחים שאלגוריתם Alpha - Beta מבצע ובנוסף מכיוון שהעץ שהוגדר בעל מקדם סיעוף קטן יותר נפתח פחות צמתים במקרה הממוצע, כלומר יהיה יותר גיזום וזמן הריצה יקטן.

**תרגיל 3 – השוואה**מבחינת זמן ריצה לסוכן Alpha - Beta ייקח פחות זמן לחשב את המהלך הבא לביצוע ולכן זמן הריצה שלנו יקטן. ניתן גם לראות זאת על ידי הגדלת העומק לחישוב ל -4 או על ידי הגדלת מספר הרוחות המשתתפות 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 ש Alpha - Beta גוזם רק ענפים שלא ישנו את ערך האסטרטגיה הסופי ולכן יוחזר אותו מהלך לביצוע בעל הערך הגדול ביותר .

חלק ה' – בניית סוכן Expectimax לרוח רנדומלית  
**תרגיל 2 – השוואה**

קיימים כמה ההבדלים בין מימוש זה לבין המימושים הקודמים:  
**הבטחת חסם תחתון** על הערך היוריסטי של האסטרטגיה עבור העומק בו אנו מסתכלים, באלגוריתמים הקודמים מובטחת לנו **שכל** סדרת מהלכים של היריב מהמהלך שבו אנו מבצעים תוביל לערך יוריסטי שהוא לפחות כמו הערך שהושג בפיתוח העץ.  
כעת האלגוריתם מסתמך על ערך תוחלת פונקציית הסתברות ולכן חסם תחתון לא מובטח לנו, מכיוון שבתורו היריב יכול לבחור במהלך אשר מקטין את ערך האסטרטגיה שחושב בפיתוח הקודם. כלומר האלגוריתם פונה למהלכים אשר ערך התוחלת שלהם הוא הגבוהה ביותר ולכן לא מבטיח ערך אסטרטגי.  
דוגמא להמחשה: ישנה רוח אחת ולה 3 מהלכים חוקיים ולפקמן 2 מהלכים חוקיים ועומק העץ הינו 2.  
כפי שניתן לראות מהגרף ערך האסטרטגיה המובטח מאלגוריתם Minimax הינו 100 כאשר הפקמן מבצע את המהלך WEST ואכן אם יבצע צעד זהה יזכה בניקוד שהינו לפחות 100 בעוד 2 צעדים. לעומת זאת ערך האסטרטגיה על פי האלגוריתם Expectimax הינו 200 כאשר יבצע הפקמן את צעד EAST אבל אם ביצע צעד זה הרוח יכולה לפנות EAST ואז הערך האסטרטגי הינו -100 כלומר האלגוריתם חזה 200 אמנם לאחר 2 מהלכים הערך הינו -100 באותה מידה גם הרוח יכולה לפנות NORTH ואז יקבל הפקמן 500.

Pacman Turn

MiniMax value:100

Expectimax value:200

EAST

WEST

Pacman Turn

heuristic value:

-100

Pacman Turn

heuristic value: 500

Pacman Turn

heuristic value:200

Pacman Turn

heuristic value:100

Pacman Turn

heuristic value:100

Pacman Turn

heuristic value:100

WEST

WEST

EAST

EAST

NORTH

NORTH

הבדל נוסף שקיים הוא דרך קבלת ההחלטות, אלגוריתם זה נבנה עבור **רוח ספציפית**, הרוח הרנדומלית, הוא מחשב את המהלך הבא לביצוע בהסתמכות על אסטרטגיית המשחק של הרוח. לעומת זאת, האלגוריתמים הקודמים לא היו מודעים לרוח שמשחקת מולם. בנוסף נשים לב כי ביוריסטיקה המשופרת שלנו יש יותר משמעות למיקום הרוח ביחס לפקמן ולכן חשוב להתחשב באסטרטגיית המשחק שלה. לכן נוכל לצפות לתוצאות טובות עבור שימוש באלגוריתם זה כאשר אנו משחקים אל מול הרוח הרנדומלית.

חלק ו' – בניית סוכן Expectimax לרוח לא רנדומלית

**תרגיל 1 – הבנת אסטרטגיית הרוח**בניית פונקציית ההתפלגות עבור המהלכים האפשריים לרוח נעשית באופן הבא:

* אוספים את כל המצבים החוקיים לביצוע.
* אם הרוח פוחדת (הפקמן יכול לאכול אותה) קובעים לה מהירות 0.5 ואחרת מהירות 1.
* מחשבים וקטורים מהמיקום הנוכחי של הרוח אל המיקומים החדשים האפשריים בהינתן המהירות.
* מוסיפים את הווקטורים על המיקום הנוכחי של הרוח.
* מחשבים מרחקי מנהטן מכל המיקומים האפשריים של הרוח אל הפקמן.
* אם הרוח פוחדת (הפקמן יכול לאכול אותה) בוחרים במרחק המקסימלי מהפקמן (הרוח רוצה להתרחק ממנו), אחרת בוחרים מהמרחק המינימלי מהפקמן (הרוח רוצה להתקרב אליו על מנת לאכול אותו).
* אוספים את כל המצבים אשר מקיימים את המרחק הטוב ביותר (רחוק או קרוב לפקמן בהתאם למצב הרוח), קובעים (זמנית) את הסתברות עבור המהלכים הכי טובים (בקוד זהו קבוע 0.8).
* לכל המצבים החוקיים האחרים קובעים הסתברות שהינה המשלים לאירוע הכי טוב.
* לבסוף מנרמלים את הערכים של כל המהלכים על מנת שתקיים תכונות פונקציה התפלגות.

דוגמא: בהינתן כי לרוח EAST וNORTH מהלכים **הכי טובים** (להתקרב או להתרחק לפקמן) וסה"כ 4 מהלכים חוקיים נקבל:



מסקנה: האסטרטגיה של הרוחות היא להעדיף את המהלכים הכי טובים לביצוע, כלומר בעת שהרוח מפוחדת להתרחק כמה שיותר מפקמן וכך להימנע מאכילה על ידי הפקמן וכאשר הרוח לא מפוחדת להתקרב כמה שיותר לפקמן על מנת לנסות לאכול אותו. ההעדפה מבוצעת על ידי מתן הסתברות גדולה הרבה יותר למצבים אלו.  
אמנם באסטרטגיה זו עדיין קיים סיכוי לבצע מהלכים חוקיים אשר אינם משפרים את מצבה של הרוח.

**תרגיל 3 – השוואה למימוש הקודם**ההבדל בין המימושים הוא שבמימוש הקודם פונקציית ההסתברות הייתה אחידה על פני מספר המהלכים החוקיים שיש ליריב לביצוע והחישוב עבורה היה מידי (בדומה להטלת קוביות). כעת פונקציית ההסתברות של היריב תלויה באסטרטגיה שלו ויש צורך לבצע לוגיקה יותר מורכבת על מנת לקבל את פונקציית ההסתברות. בנוסף מכיוון **שקוד סוכן הרוח חשוף** עבורנו אנו יכולים לחשוף את הסוכן שלנו לאסטרטגיה זו אך אם לא היה חשוף היינו צריכים לנחש או ללמוד את האסטרטגיה שלו.

**תרגיל 4 – שיפור אסטרטגיית הרוח**

1. כאשר ישנם מהלכים הכי טובים לביצוע, כלומר להתקרב או להתרחק מהפקמן בהתאם למצב הרוח ניתן הסתברות רק למצבים אלו ולא לכל המצבים חוקיים. באסטרטגיה זו הרוח ממש מנסה להתקרב או להתרחק מהפקמן בהתאם למצבה ולא מבצעת מצבים שיכולים להרוס לה.
2. האסטרטגיה של רוח בודדת כעת לא מודעת לשאר הרוחות המשחקות ולכן נרצה להוסיף את הידיעה הזאת כאשר כל רוח תנסה להתרחק משאר הרוחות ובצורה זו הרוחות ינסו לרדוף אחרי הפקמן אבל במסלולים שונים ולכן ינסו לאכול את הפקמן או לברוח ממנו בכיוונים שונים וכך יהיה לו יותר קשה לברוח מהן או לתפוס אותן.

חלק ז' – ניסוח השערות במשחק פקמן  
**תרגיל 1 – ניסוח השערות ומבחן**

נבנה את התצפית על ידי הרצה של שני האלגוריתמים על פני מספר רב משחקים עם לוחות שונים ונאסוף את הניקוד שהושג בכל משחק.

נשתמש במבחן ה- T-Test על מנת לבחון האם לדחות או לקבל את השערת האפס. כאשר עבור המבחן נסמן:



בעזרת ערך זה ניגש ל T-Test table עם דרגת חופש :  ונבדוק האם לדחות את השערת האפס או לא לפי ה- T- value שחושב.   
(מהסתכלות באינטרנט זאת השיטה לדחות/ לקבל את השערת האפס במבחן T-Test).

חלק ח' – ניסויים, תוצאות ומסקנות.

**תרגיל 2 – גרף ניקוד ממוצע בתלות בעומק החישוב**

**תרגיל 3 – מסקנות מהגרף**ניתן להסיק מהגרף את המסקנות הבאות:

* היוריסטיקה החדשה שמימשנו טובה משמעותית מהיוריסטיקה הפשוטה שסופקה לנו, ניתן לראות זאת על ידי השוואת בין שני הסוכנים Reflex Agent ו Better Agent, כפי שצפינו.
* אין הבדל משמעותי בין ממוצע הנקודות משיג הסוכן Alpha Beta Agent לבין הסוכן Minimax, כפי שצפינו ההבדל ביניהם הוא רק בזמן ריצה, בזמן חישוב מהלך הבא לביצוע.
* ישנה מגמת עלייה בממוצע הניקוד שהסוכנים משיגים ככל שמחשבים יותר עמוק, צפינו זאת מכיוון שבחירת המהלך הבא לביצוע מושפע מיותר מהלכים קדימה ולכן בחירתו טובה יותר.
* אלגוריתם Random Expectimax משיג תוצאות טובות יותר משאר האלגוריתם, כפי שצפינו זאת מכיוון שאלגוריתם נבנה במפורש עבור הרוח הרנדומליות ומודע לאסטרטגיית המשחק שלה, בנוסף אלגוריתם זה פחות שמרני בבחירות המהלכים ולוקח יותר סיכונים ולכן יש לו יותר פוטנציאל להגיע לניקוד גבוה יותר.

**תרגיל 4 – גרף זמן חישוב התור בתלות בעומק החישוב**

**תרגיל 5 – מסקנות מהגרף**ניתן להסיק מהגרף את המסקנות הבאות:

* זמן חישוב היוריסטיקה החדשה גבוה מזמן חישוב היוריסטיקה הפשוטה שסופקה לנו. כפי שצפינו המצאנו יוריסטיקה מסובכת ולכן זמן החישוב שלה גדול משמעותית מהיוריסטיקה הפשוטה שסופקה לנו, ניתן לראות זאת על ידי השוואת בין שני הסוכנים Reflex Agent ו Better Agent.
* ככל שמגדלים את עומק עץ ה – Minimax זמן חישוב המהלך הבא לביצוע גדל אקספוננציאלי, כפי שצפינו זאת מכיוון שמספר העלים אשר מוערכים יוריסטית גדל אקספוננציאלי.
* אלגוריתם Random Expectimax ואלגוריתם Minimax בעלי אותו זמן חישוב תור, כפי שצפינו זאת מכיוון ששיניהם לא מבצעים גיזום וההבדל היחיד ביניהם הוא שהאחד מחשב תוחלת על הערכים והשני מחשב מקסימום ולכן בעלי זמן חישוב כמעט זהה.
* Alpha Beta Agent בעל זמן חישוב קטן משמעותית מאלגוריתם Minimax, כפי שצפינו בגלל הגיזום המתבצע נחסך הרבה זמן חישוב.
* ככל שהעומק גדל ההפרש בין זמן החישוב של Alpha Beta Agent לבין זמן החישוב של אלגוריתם Minimax גדל, כפי שצפינו בגלל שמספר הפיתוחים הממוצע תלוי בעומק העץ עליו מחשבים את האסטרטגיה.

**תרגיל 6 –Random Expectimax Vs. Directional Expectimax**

|  |  |  |
| --- | --- | --- |
| Agent/Ghost | Directional Ghost [Average Score] | Random Ghost [Average Score] |
| Random Expectimax Agent | 1263.8 | 2337.2 |
| Directional Expectimax Agent | 1114.8 | 2119.4 |

מסקנות:

* שני האלגוריתמים השיגו תוצאות טובות יותר כאשר שיחקו מול הרוח הרנדומלית (ביחס לרוח המכוונת), כפי שציפינו מכיוון שרוח זו אינה מודעת לקיומו של הפקמן ואינה מנסה לרדוף אחריו ולכן לסוכן הפקמן סיכוי גדול יותר לנצח את המשחק (גם כאשר הוא חושב שהיא רודפת אחריו).
* אלגוריתם Random Expectimax שיחק טוב יותר מול הרוח הרנדומלית מאשר אלגוריתם Directional Expectimax, כפי שצפינו מכיוון שאלגוריתם זה נבנה בהסתמך על אסטרטגיית הרוח הרנדומלית לעומת האלגוריתם Directional Expectimax אשר אינה מודע לאסטרטגיית משחק שונה משלה.
* אלגוריתם Random Expectimax השיג יותר ניקוד מאשר אלגוריתם Directional Expectimax כאשר שיחק מול הרוח המכוונת בלוח המבוקש. תוצאה זו אינה תואמת את הציפיות שלנו כי ציפינו שסוכן זה ישיג פחות ניקוד ויפסל מהר יותר מאשר הסוכן המודע לאסטרטגיית הרוח המכוונת. תוצאה זו אינה חד משמעית כי ההבדל בין התוצאות קטן ולכן נצטרך לדגום יותר משחקים על מנת לברר תוצאה זו.

**תרגיל 7 –Minimax Classic Layout, depth is 4**

אפיון הלוח: לוח זה מאוד קטן וצפוף ולכן ישנה חשיבות גדולה יותר להסתכלות לעומק עץ המצבים על מנת לחשב את המצב הבא לביצוע. זאת מכיוון שהפקמן יכול להפסיד תוך מספר קטן של מהלכים ולכן כאשר מסתכל עמוק יותר בעץ הפקמן יוכל לתכנן מסלול שהוא לא מפסיד בו החל מהמהלך הראשון. (ראינו כי אם לא מוצא כזה מעדיף להיפסל מאשר להפסיד עוד נקודות).

|  |  |
| --- | --- |
| Agent/Ghost | Random Ghost [Average Score] |
| Reflax Agent | -205.15 (d=1) |
| Better Agent | 225.86 (d=1) |
| Minimax Agent | 513 |
| Alpha-Beta Agent | 363.857 |
| Random Expectimax Agent | 370.57 |

מהתוצאות בטבלה ניתן להסיק כי אלגוריתמים אשר מסתכלים לעומק העץ משיגים תוצאה טובה יותר בלוח זה.  
נשים לב כי אלגוריתם Minimax לעומת Random Expectimax הניב תוצאה טובה יותר זאת מכיוון שהינו אלגוריתם שמרני שתמיד מניח את המקרה הגרוע ביותר כאשר הרוח משחקת. לעומתו אלגוריתם Random Expectimax פחות שמרני ונוטל יותר סיכונים ולכן בלוח קטן וצפוף שכזה מניב תוצאה פחותה מכיוון שהסיכונים בלוח זה גדולים לעומת הלוחות גדולים והפחות צפופים.

**תרגיל 8 –Trapped Classic Layout, depth is 4**

אפיון הלוח: לוח זה מאוד קטן, צפוף וישנו מסלול יחיד עבור סוכן הפקמן לנצח את המשחק. בנוסף, מסלול יחיד זה תלוי בהתנהגות הרוח. רק במידה שהרוח שחוסמת את האוכל "מפנה" את הדרך יש אפשרות לנצח את המשחק.

|  |  |
| --- | --- |
| Agent/Ghost | Random Ghost [Average Score] |
| Reflax Agent | -501.57 (d=1) |
| Better Agent | -206.57 (d=1) |
| Minimax Agent | -501 |
| Alpha-Beta Agent | -501 |
| Random Expectimax Agent | -58.857 |

מהתוצאות בטבלה ניתן להסיק כי אלגוריתמים Minimax משיגים תוצאה פחות טובה יותר בלוח זה. מסקנה זאת נובעת מהעבודה שאלגוריתם Minimax קובע את ערך האסטרטגיה לפי המקרה הגרוע ביותר (שבו הרוח שועטת לכיוון פקמן) ואילו אלגוריתם Random Expectimax לוקח סיכונים ומנסה להשיג כמה שיותר נקודות. במפה זו, התנהגות קונסרבטיבית תוביל להחלטה של הפסד מיידי מכוון ע"י פקמן, זאת כדי שפקמן יפסיד כמה שפחות נקודות. מצד שני, הימור על ניקוד גבוה יותר, עלול להביא להפסד עם תוצאה גרועה יותר אך רק בעזרתו ניתן לנצח את המשחק ולהשיג ניקוד גבוה יותר משמעותית. למען האמת, לעומק העץ כמעט אין משמעות בלוח זה.

**תרגיל 9 – ביצועים וניתוח תוצאות**  
כחלק מניתוח תוצאות המשחקים (קובץ *csv*) ביצענו השוואה בין ביצועי האלגוריתמים כתלות בעומק עץ החישוב וכתלות בלוח המשחק. השוואה זו הניבה לנו את היכולת לנתח כל סוכן ביחס לאחר ובנוסף לנתח את התנהגות הסוכנים בתלות בלוח המשחק.  
נתחיל מניתוח הסוכן שסופק לנו, הסוכן Reflex Agent (משתמש ביוריסטיקה שסופקה לנו) הינו הסוכן בעל הביצועים הגרועים ביותר, ללא תלות בעומק החישוב של שאר האלגוריתמים או בלוח המשחק. זאת ככל הנראה כתוצאה משימוש ביוריסטיקה פשוטה מידי להערכה מצב במשחק. לכן, היוריסטיקה המשופרת שבנינו גוברת על היוריסטיקה הפשוטה.  
לעומתו הסוכןBetter Agent משיג ביצועים טובים יותר מאלגוריתמי Alpha-Beta ו Minimax בלוחות משחק קטנים, כך למשל בלוחות: OpenClassic, testClassic, trappedClassic הוא משיג תוצאות טובות יותר מהם בכל עומק שחושב. הסיבה האפשרית לכך היא מכיוון שבלוחות קטנים מאוד תלויים בתזוזה של הרוח ואלו אלגוריתמים שמרניים ולא לוקחים סיכונים וכנראה יעדיפו להיפסל מוקדם יותר מאשר להפסיד ניקוד בהמשך. לעומתם סוכן אשר מחשב רק צעד אחד קדימה אינו מודע שיכול להיפסל בעוד כמה מהלכים ולכן מבצע מהלך אשר נוטל סיכון גם בלא ידיעתו.  
כאשר אנחנו מסתכלים על כל אחד ביצועי כל סוכן (המשתמשים בעומק) כתלות לעומק שהשתמש וללוח אנו מגלים כי הגדלת העומק אינו תמיד משפרת את ביצועי הסוכן וכי הדבר תלוי בלוח המשחק. דווקא בלוחות הגדולים הגדלת העומק לחישוב אינה משפרת באופן חד משמעי את ביצועי הסוכן ואילו בלוחות הקטנים ישנו שיפור שניתן להבחנה זאת מכיוון שבלוח קטן תכנון המהלך הבא בהסתמך על מהלכים עתידיים של שאר הסוכנים יכול להבטיח היחלצות מהפסד, או לאסוף יותר נקודות. לכן לא תמיד עדיף להגדיל את עומק העץ ולפגוע בביצועים על ידי הגדלת זמן חישוב המהלך הבא לביצוע.  
כפי שניתן היה לצפות מראש הסוכנים Alpha- Beta ו Minimax בעלי ביצועים דומים ובעלי התנהגות דומה אל מול סוכן Random Expectimax.  
הסוכן