דו"ח מסכם

תרגיל בית 2­

יונתן בתן

302279138

[Yonibettan@gmail.com](mailto:Yonibettan@gmail.com)

עמרי פרוינד

301695490

omrifro[@gamil.com](mailto:Neliahu@gamil.com)

# **חלק א'**

# סעיף 1:

בקוד

# סעיף 2:

בקוד

# סעיף 3:

בהינתן gameState מסוים הסוכן ReflexPlayer מחשב את כל הצעדים האפשריים שלו, כלומר מתוך הקבוצה , לאחר מכן מחשב את הערך של צעד על המצב הנתון לכל צעד אפשרי באמצעות היוריסטיקה ולבסוף בוחר בצורה רנדומלית צעד מבין הצעדים המניבים את הערך המקסימלי.

ההיוריסטיקה בה הוא משתמש מחזירה את ה- score הנתון למצב עליה היא פועלת, score זה הוא הניקוד הנוכחי ב-state כפי שהוא מופיע ב- GUI.

נבחין כי היוריסטיקה זו מניבה סוכן הבוחר את הפעולה אשר מניבה ניקוד מקסימלי מבלי לחשוב קדימה ולנסות מצבים שאינם אופטימליים לצעד הבא אבל יניבו תוצאה טובה יותר בהמשך.

# **חלק ב'**

# סעיף 1:



* הפרמטר מייצג את הניקוד הנוכחי של המצב כפי שהוא מופיע ב- GUI
* הפרמטר מייצג את המרחק לחתיכת אוכל הקרובה ביותר
* הפרמטר מייצג את המרחק לקפסולה הקרובה ביותר
* הפרמטר הינו חיובי אם פקמן אכל קפסולה ושלילי אחרת
* הפרמטר מייצג את המרחק לרוח הקרובה ביותר
* הפרמטר מוסיף 1 לערך היוריסטי בהסתברות

# סעיף 2:

תחילה נבחין כי הפרמטר מופיע בחישוב התוצאה ושאר המספרים הינם מספרים הקטנים או שווים ל-1 ולכן מבצעים "fine tuning" להיוריסטיקה scoreEvaluationFunction.

* כפי שלמדנו, עבור מצבים סופיים ההיוריסטיקה מחזירה את הערך שמחזירה פונקציית התועלת
* הפרמטר מייצג את ניקוד המצב הנוכחי ולכן הוא פרמטר חשוב בפונקציה שלנו שכן מצב של ניצחון מכיל כ-500 נקודות יותר מקודמו ולכן נרצה לתת לכך משקל רב בערך היוריסטי.
* לכל אחד מהפרמטרים נרצה לתת בונוס גבוהה כאשר המרחק קטן ובונוס נמוך כאשר המרחק גדל ובנוסף עבור מרחקים גדולים ההבדל בין מצבים שונים נהיה זניח מבחינת פרמטר זה ולכן בחרנו בפונקציה .
* עבור המרחק מהרוח המינימלית אנו מבדילים בין המצב בו פקמן אכל קפסולה ולכן הרוח פגיעה למקרה בו פקמן לא אכל קפסולה.

במקרה הראשון נרצה להתקרב לרוח על מנת לאכול אותה ולצבור נקודות ומיכון שאכילת רוח מזכה בפי 20 יותר נקודות מאשר חתיכת אוכל רצוי לתת פקטור של 20 אל מול הבונוסים של חתיכות האוכל, לאחר משחק עם הפרמטרים גילינו שפקטור 30 נותן את התוצאה הטובה ביותר

* הפרמטר נועד למנוע היתקעות במינימום מקומי, כלומר כאשר ישנם 2 מצבים סמוכים כך שבמעבר לאחד מהם השני נהיה הטוב ביותר אנחנו ניתקע והאלגוריתם לא יסיים את ריצתו ולכן תוספת של רנדומליות מאפשרת לנו לצאת ממצבים אלו
* נשים לב כי ללא הפרמטר לא היינו צריכים להפריד בין מצבים סופיים לכאלו שאינם סופיים שכן כל ערכי המינימום מאותחלים ל- ולכן עבור מצבים סופיים ערכים אלו לא מתעדכנים והיינו מקבלים

# סעיף 3:

בקוד

# **חלק ג'**

# סעיף 1:

ההנחה שלנו הינה שהסוכנים משחקים בזה אחר זה במשחק ובפרט פקמן משחק בהתאם לצעד האחרון של רוח מס' 1 שהיא משחקת בהתאם לצעד האחרון של רוח מס' 2 וכ"ו.

הנחה זו כמובן אינה נכונה שכן הרוחות אינן בהכרח מגיבות אחת לשנייה, יתכן והן פועלות באופן בלתי תלוי אחת מהשנייה ואם במקרה הן כן מגיבות אחת לשנייה לא ניתן להבטיח שרוח מס' 1 תגיב דווקא לרוח מס' 2 ולא להפך למשל.

# סעיף 2:

בקוד

# סעיף 3:

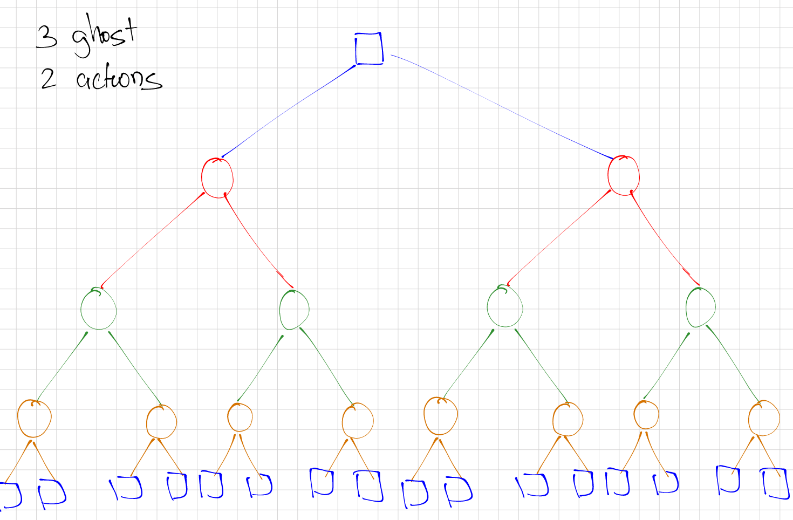
לא בטוח בפתרון

במימוש הנוכחי, צומת בשכבת מינימום מייצג את הפעולה של הרוח ה- בלבד על המכיל בתוכו כבר את הפעולות של הרוח ה- לכל באותו התור.

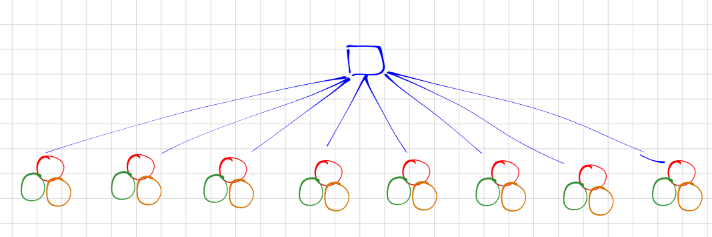
במימוש החדש תהיה לנו שכבת מינימום יחידה אשר כל מצב בה מייצג את פעולות כל הרוחות בתור מסוים.

יתרונות:

* ניתן להשתמש באלגוריתם מינימקס פשוט של 2 שחקנים אותו למדנו
* יש צורך לבצע פחות חישובים של מינימקס כי יש פחות מצבים אפשריים בשכבת המינימום מאשר בסכום שכבות המינימום
* נניח שיש לנו רוחות וכל סוכן יכול לבצע פעולות ואנו רוצים לחשב מינימקס עם הגבלת עומק
* במימוש הנוכחי נצטרך לחשב ערכי מינימקס.



* במימוש החדש יש לנו רק מצבים בשכבת המינימום בעומק (כל מצב מייצג מסלול מרוח אדומה לרוח כתומה) ולכן צריך לחשב ערכי מינימקס



חסרונות:

* מורכב יותר למימוש
* חישוב מצבי הבנים דורש בניית מצב לכל קומבינציה אפשרית של פעולות הרוחות וזה מורכב יותר מאשר ליצור בן עבור כל פעולה של רוח יחידה כפי שעשינו.

# **חלק ד'**

# סעיף 1:

לא, רק פקמן נחשב כצומת מקסימום ולכן נכנס לחלק הראשון של התנאי ואילו כל שאר הרוחות נחשבות לצמתי מינימום ולכן נכנסים לחלק השני של התנאי באלגוריתם.

* גיזום :
* בפיתוח צומת מינימום , אם אחד הבנים שלו מניב ערך קטן מ- אזי מובטח כי ולכן ענף זה ייגזם בדיוק כמו במבנה עץ המכיל שכבת מינימום יחידה לכל שכבת מקסימום.
* גיזום :
* בפיתוח צומת מקסימום , אם אחד הבנים שלו מניב ערך גדול מ- אזי מובטח כי ולכן ענף זה ייגזם בדיוק כמו במבנה עץ המכיל שכבת מינימום יחידה לכל שכבת מקסימום.

סה"כ אופן פעולת הגיזום אינו משתנה ולכן מבנה העץ החדש שהגדרנו אינו משפיע על האלגוריתם אלפא-ביתא.

# סעיף 2:

בקוד

# סעיף 3:

כן.

כיוון שאנו ממיניים את הבנים בעזרת ההיוריסטיקה אנו בעצם מתקרבים לגיזום האופטימלי (במקרה של היוריסטיקה מושלמת היינו מקבלים גיזום אופטימלי) ולכן יתבצעו יותר גיזומים, משמע פחות פיתוחים ולכן זמן חישוב קצר יותר.

אם בסוף השארתי את ההיוריסטיקה לא דטרמניסטית אז לעדכן את התשובה

ברמה העקרונית לא, אך בפועל יתכנו מהלכים שונים.

"ברמה העקרונית" משמעותו שאם מדובר תיאורטית באותה ריצה, כלומר תנועת הרוחות זהה ואנו בודקים רק את התנהגות הסוכנים, ובנוסף ערכי המינימקס של הבנים הינה סדרה מונוטונית עולה ממש (אין 2 בנים עם ערך מינימקס מקסימלי זהה) אזי במקרה זה בחירת הצעדים של הסוכנים תהיה זהה.

לפי משפט הנכונות של אלפא-ביתא האלגוריתם מחזיר את ערך המינימקס של תת העץ שלו ולכן מובטח שהערך של כל אחד מהבנים יהיה ערך המינימקס שלו ולכן לכאורה אין שוני בין המהלכים שבוחרים הסוכנים.

כיוון שעבור בנים שונים המניבים ערך מינימקס מקסימלי זהה אלגוריתמי בחירת הצעד בוחרים צעד באופן רנדומלי (כך מימשנו את אלגוריתם בחירת הצעד) לכן סביר שבמקרה בו יש יותר מבן יחיד המניב ערך מינימקס מקסימלי 2 הסוכנים יפלו על ערך רנדומלי שונה ויבחרו בצעד שונה.

בנוסף אם מריצים את 2 הסוכנים (ריצות שונות) יתכנו תנועות שונות של הרוחות שכן איננו רשאים להניח דבר על התפלגות התנועה של הרוחות וכיוון שההיוריסטיקה שלנו מתחשבת במיקום הרוחות אזי גם במקרה זה יתכנו בחירת מהלכים שונים בין הסוכנים.

# **חלק ה'**

# סעיף 1:

בקוד

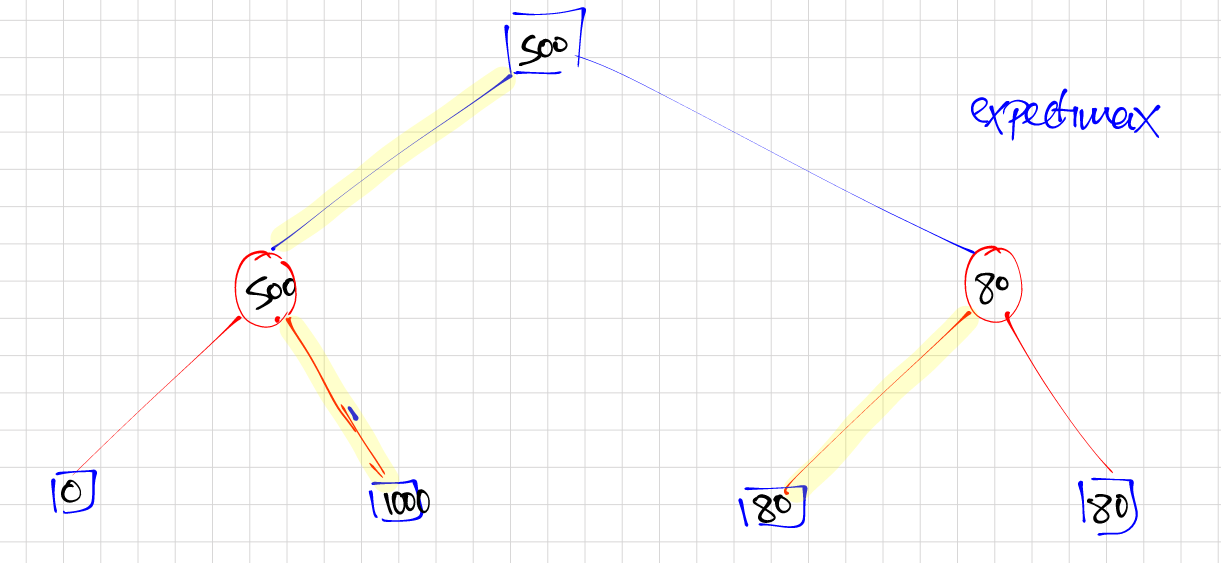
# סעיף 2:

סוכנים המשתמשים באסטרטגיית מינימקס מניחים שהיריב בוחר בצעד הגרוע ביותר עבורנו, כלומר היריב עושה הכל על מנת לפגוע בנו ככל הניתן.

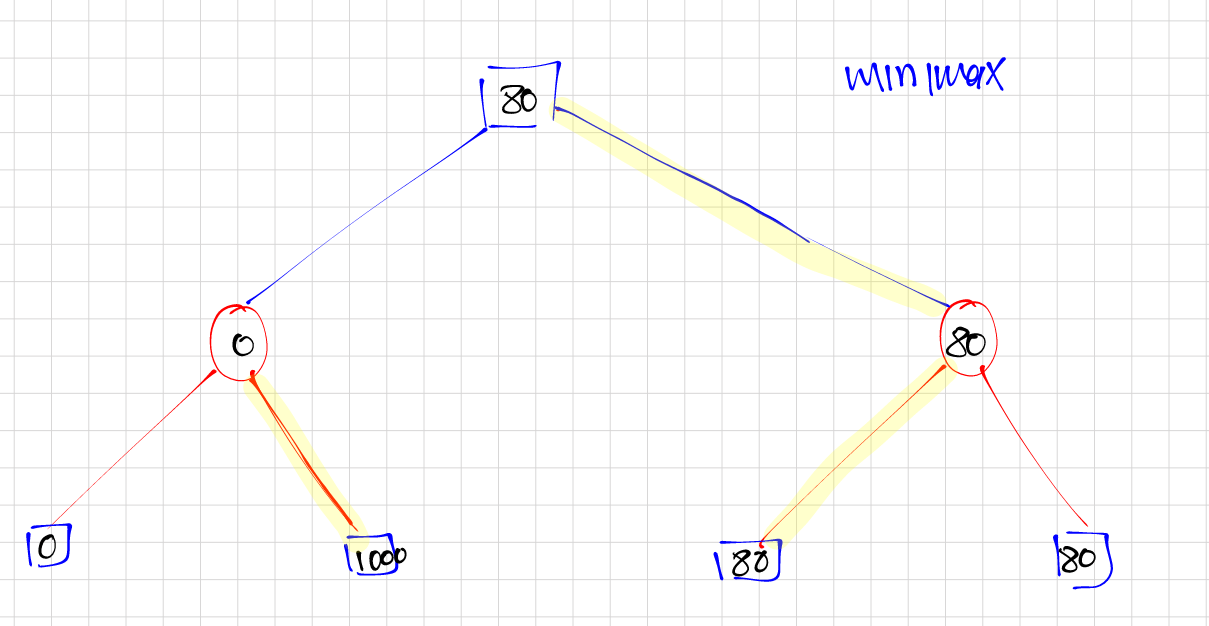
סוכן אקספקטימקס לעומת זאת חושב בצורה רציונלית יותר עבור משחקים הסתברותיים, בסעיף זה אנו מניחים שהרוח בוחרת את הצעד הבא בהתפלגות אחידה על מרחב הצעדים האפשריים שלה, כלומר צמתי הרוחות הם צמתים הסתברותיים ולכן יותר מציאותי לנבא את הצעד של הרוח כתוחלת של אפשרויות התנועה שלה בהתפלגות אחידה מאשר כצעד שיפגע בנו בצורה המקסימלית.

אנו מאמינים כי שימוש באקספקטימקס עבור רוחות רנדומליות ייתן תוצאות טובות יותר שכן מינימקס נותן חסם תחתון לפתרון האופטימלי אבל בהינתן שהרוח רנדומלית יתכן וקיימים פתרונות טובים יותר שאותם אנו עלולים לפספס עם מינימקס\אלפא-ביתא.

לדוגמא עבור רוח יחידה, נניח שהרוח בוחרת את הצעד המסומן אזי אקספקטימקס בוחר את הצעד המוביל לתוצאה המקסימלית במשחק.



ואילו מינימקס היה בוחר את הצעד השני אשר מניב תוצאה טובה הרבה פחות.



# **חלק ו'**

# סעיף 1:

* התפלגות התנועה:

ההתפלגות תלויה בפרמטר הסתברותי .

נסמן את מספר הפעולות החוקיות ב- ואת מספר הפעולות אשר מניבות ערך אופטימלי ב- .

הרוח בוחרת בצעד אופטימלי (מתקרבת לפקמן אם אינה פגיעה ומתרחקת אחרת) בהסתברות

ובצעד שאינו אופטימלי בהסתברות

.

כפי שניתן לראות אין העדפה לצעד מסוים בתוך קבוצת הצעדים האופטימליים, כנ"ל עבור הצעדים הלא אופטימליים.

לצורך הבהרה נראה מה קורה בערכי הקצה:

* עבור נקבל

כלומר הסוכן בוחר צעד בהתפלגות אחידה מבין הצעדים האופטימליים.

* עבור נקבל

כלומר הסוכן בוחר צעד בהתפלגות אחידה מבין כל הצעדים האפשריים.

כברירת מחדל מתקיים ובכל מקום בקוד השימוש ברוח הנ"ל היא תמיד עם ערך ברירת המחדל.

* האסטרטגיה:

הרוח בוחרת בצורה רנדומלית צעד מבין הצעדים האופטימליים בהסתברות גבוהה ובוחרת בצורה רנדומלית צעד מבין הצעדים שאינם אופטימליים בהסתברות נמוכה.

צעד אופטימלי הוא צעד אשר מקרב את הרוח לפקמן באופן מקסימלי כאשר הרוח אינה פגיעה, כלומר במקרים בו פקמן לא אכל קפסולה או שחלף מספיק זמן מאז, או צעד אשר מרחיק את הרוח מפקמן באופן מקסימלי כאשר הרוח כן פגיעה.

# סעיף 2:

בקוד

# סעיף 3:

המימוש שלנו כמעט זהה בין 2 הסוכנים, לכל אחד מהם יש מתודה יחידה אשר מחזירה את הצעד הבא, ונעזרת במתודה rbExpectimax לצורך חישוב הערך של צמתי מינימום.

ההבדל היחיד הוא ש- RandomExpectimaxAgent משתמש במתודה זו עם הפרמטר ‘random\_ghost’ ואילו DirectionalExpectimaxAgent משתמש במתודה עם פרמטר ‘directional\_ghost’.

מתודה זו מתנהגת כמו מינימקס עבור צמתי מקסימום ומחשבת את ערך התוחלת עבור צמתי מינימום כתלות בהתפלגות התנועה של טיפוס הרוח אשר מתקבל כארגומנט.

החישוב מתבצע לפי הפילוג אשר הוסבר בסעיף 1 שאל חלק זה.

# סעיף 4:

* שיפור 1:

נחשב את המרחק מהרוח לפקמן תוך התחשבות במרחק האמתי למשל ע"י שימוש ב- GameState.getWall() במקום מרחק מהנהטן שהרי יתכן שרוח תצטרך "לחזור אחורה" על מנת לעבור קיר מסוים ומרחק מנהטן מפספס עניין זה.

* שיפור 2:

תוצאה אופטימלית לא תהיה תלויה אך ורק במרחק לפקמן אלא גם תיקח בחשבון את המרחק לחתיכות אוכל וקפסולות בידיעה שפקמן רוצה להגיע למקומות אלו ולכן תוכל לנצל זאת לטובתה.

# **חלק ז'**

# **חלק ח'**

# **חלק ט'**