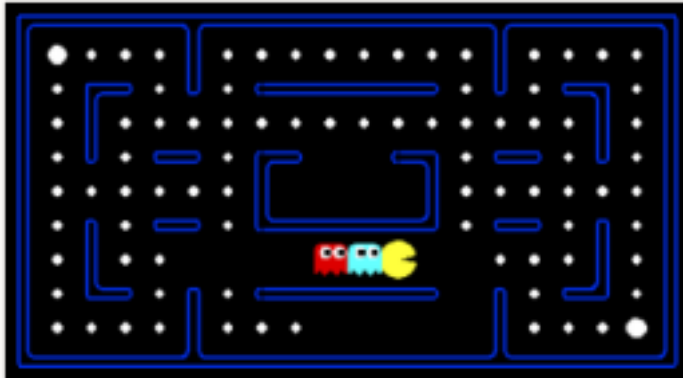


Pacman



הקדמה:

בפרויקט זה תבנו סוכנים לגרסה הקלאסית של פקמן, כולל רוחות רפאים. לאורך הדרך גם תיישמו α -Beta Pruning ו- minmax.

בסיס הקוד לא השתנה הרבה מהפרויקט הקודם, אך אנא הורידו את הקבצים מאזור classroom ב zip תרגיל 2 הנמצאים בתיקיית במקום לשלב קבצים מהפרויקט הקודם.

קבצים לעריכה:

הקובץ בו תממשו את הסוכנים שלכם – `py.multiAgents`

קבצים שכדאי להסתכל עליהם:

`Pacman GameState` הקובץ הראשי שמריץ משחקי פאקמן. קובץ זה מתאר את ה- `py.pacman` שבו אתה משתמש בפרויקט, זה.

לוגיקת המשחק של עולם הפקמן. קובץ זה מתאר מספר טיפוסים תומכים כמו – `py.game` `Grid`, `AgentState`, `Agent`, `Direction`.

מבני נתונים שימושיים ליישום אלגוריתמי חיפוש – `py.util`

קבצי תמיכה שניתן להתעלם מהם:

`graphicsDisplay.py` - גרפיקה

`graphicsUtils.py` - בגרפיקת תמיכה Pacman

עבור פאקמן ASCII גרפיקת - `py.textDisplay`

סוכנים לשליטה ברוחות רפאים - `py.ghostAgents`

ממשקי מקלדת לשליטה בפקמן - `py.keyboardAgents`

קוד לקריאת קבצי פריסה ואחסון תוכנם - `py.layout`

הגשה

עליכם לשלוח קבצים אלה עם הקוד והערות. `py.multiAgents` במהלך התרגיל תערכו חלקים של שלכם. נא לא לשנות את הקבצים האחרים או לשלוח אף אחד מהקבצים המקוריים מלבד קובץ זה. בראש הקובץ נא לכתוב את שם הסטודנט ות.

והוא יכיל את שם `txt.detail` בנוסף לקובץ זה, עליכם להגיש קובץ פרטים אישיים בשם

ה
o
ט

ו
ד
נ
ט
ב
ש
ו
ר
ה
ה
ר
א
ש
ו
נ
ה
ו
ת
.
ז
.
ב
ש
ו
ר
ה
ה
ש
נ
י
ה
.

Pacman Agent-Multi! ברוך הבא ל

קלאסי על ידי הרצת הפקודה pacman ראשית, תוכלו לשחק

```
python pacman.py
```

תוך שימוש בחיצי המקלדת לניווט.

py.multiAgents שמסופק עבורכם ב ReflexAgent כעת, הריצו את הסוכן

```
python pacman.py -p ReflexAgent
```

שימו לב שהוא משחק מאוד גרוע אפילו בפריסות פשוטות

```
python pacman.py -p ReflexAgent -l testClassic
```

וודאו שאתם מבינים את צורת הפעולה שלו היטב, py.multiAgents שב ReflexAgent עיינו בקוד של

שאלה 1 - Reflex Agent

כך שישחק טוב יותר. קוד הסוכן שקיבלתם מספק כמה py.multiAgents שב ReflexAgent שפרו את חייב להתחשב גם במיקומי המזון ReflexAgent ה. GameState דוגמאות לפו' שמבקשות מידע מה וגם במיקומי הרוחות רפאים כדי להשיג ביצועים טובים יותר. על הסוכן שלכם לאכול את כל ה"פרסים" בקלות מהלוח הבא

```
python pacman.py -p ReflexAgent -l testClassic
```

על הלוח הבא, עם רוח רפאים אחת או שתיים (וללא אנימציה כדי להאיץ ReflexAgent נסו את

את
התצוג
ה)

```
python pacman.py --frameTime 0 -p ReflexAgent -k 1
```

```
python pacman.py --frameTime 0 -p ReflexAgent -k 2
```

כיצד הסוכן הסתדר? סביר להניח שעם שתי רוחות רפאים הוא ימות, אלא אם כן פו' ההערכה

ש
ל
ך
מ
ס
פ
י
ק
ט
ו

ב
ה

טיפים:

asList() יש את הפונקציה newFood – זכרו של .
נסו להשתמש בתכונות על ערכים (כמו מרחק לאוכל) ולא להשתמש רק בערכים עצמם. נסו .
ניתן לעשות זאת על ידי debugging להסתכל בתוכן הפנימי של האובייקטים השונים ב
על ידי newGhostStates הדפסת יצוג המחרוזות של האובייקטים. לדוגמה אפשר להדפיס
הפקודה: print(newGhostStates).

הצעה: רוחות ברירת המחדל הן רנדומליות; אפשר לשחק בשביל הכיף עם "רוחות כיוון" חכמות
יותר אם
תכתוב:

DirectionalGhost -g

אם הרנדומליות מונעת מכם לדעת האם הסוכן משתפר, תוכלו להריץ את המשחק עם
f. (רנדומליות קבועה) אותן בחירות אקראיות לאורך כל המשחק בכל משחק (על ידי הוספת
כדי להריץ הרבה – q או לכבות את הגרפיקה עם – n ניתן גם לשחק מספר משחקים ברצף עם
משחקים במהירות

שאלה 2 – Minimax

שנמצאת בקובץ MinmaxAgent בשאלה זו תממשו סוכן חיפוש יריב במחלקה המסופקת לכם
שלך צריך לעבוד עם כל כמות של רוחות רפאים, ולשם כך Minmax סוכן ה. py.multiAgents
שלך יהיו Minmax תצטרך לכתוב אלגוריתם קצת יותר כללי ממה שראית בהרצאה. בפרט, לעץ ה
(pacman של) max אחת לכל רוח רפאים (עבור כל שכבת min מספר שכבות

שמתקבל בשורת depth הקוד שלכם צריך להרחיב גם את עץ המשחק לעומק שרירותי, (משתנה
שמוגדרת כברירת defaultFunction.evaluation (כשאת ניקוד העלים תחשב בעזרת הפונקציה
scoreEvaluationFunction מחדל ל

ול depth.self שנותנת גישה ל MultiAgentSearchAgent מרחיבה את MinmaxAgent

self.evaluationFunction.

שלכם מתייחס לשני המשתנים האלו במידת הצורך מכיוון שהערכים Minmax תוודאו שקוד ה
שלכם עשויים להשתנות בתגובה לאפשרויות שורת
הפקודה.

ושל כל תגובת הרוחות, כך pacman שימו לב: שכבת חיפוש אחת נחשבת למהלך אחד של
וכל רוח רפאים pacman שחיפוש בעומק 2 יכלול את
שזזה פעמיים.

ישפיעו על generateSuccessor.GameState שימו לב 2: קריאות מיותרות או חסרות ל

הציון.

טיפי

ים:

ממשו אלגוריתם רקורסיבי תוך שימוש בפונקציה עזר .

- יפסיד בחלק מהמשחקים. זו לא בעיה מכיוון pacman יוביל לכך ש minmax ישום נכון של . שזו התנהגות נכונה ותעבור את הבדיקות
- אין). (evaluationFunction.self בחלק זה כבר כתובה Pacman פונקציית ההערכה למבחן . לשנות את הפונקציה הזו, אך שימו לב שכעת אנו בוחנים מצבים ולא פעולות, כפי שהיה ב העריך פעולות מהמצב ReflexAgent הסוכן שלך מעריך מצבים עתידיים ואילו. ReflexAgent הנוכחי
- הם 9, 8, -7, 492 לעומקים 1, 2, minimaxClassic ערכי המינימקס של המצב ההתחלתי ב . 3 ו 4- בהתאמה. שימו לב כי סוכן המינימקס שלכם יזכה לעתים קרובות (1000/665 משחקים עבורנו) למרות הציון הצפוי הנמוך של מינימקס בעומק 4

```
python pacman.py -p MinimaxAgent -l minimaxClassic -a depth=4
```

- פקמן הוא תמיד סוכן 0, רוחות הרפאים מקבלים מספור עולה לפי סדר .
- או להפיק getAction -להעביר אותן ל, GameStates כל המצבים במינימקס צריכים להגיע מ . generateSuccessor.GameState אותן באמצעות
- ברירת המחדל(, תמצאו את openClassic ו- mediumClassic על לוחות גדולים יותר כגון . פאקמן טוב ב"לא למות", אבל גרוע בניצחון. לעתים קרובות הוא יסתובב בלי להתקדם. הוא עשוי אפילו להסתובב ממש ליד נקודה מבלי לאכול אותה כיוון שהוא לא יודע לאן הוא יגיע אחרי שיאכל את הנקודה הזו, התנהגות שכזו תקינה
- כאשר פאקמן מאמין שמותו בלתי נמנע, הוא ינסה לסיים את המשחק בהקדם האפשרי בגלל . העונש המתמיד על החיים (ירידת ניקוד ככל שהמשחק מתקדם). לפעמים, זה לא הדבר הנכון לעשות עם רוחות רנדומליות אקראיות, אך סוכני מינימקס תמיד מניחים את הגרוע ביותר

```
python pacman.py -p MinimaxAgent -l trappedClassic -a depth=3
```

וודאו שאתה מבין מדוע פאקמן ממחר לרוח הרפאים הקרובה ביותר במקרה זה

שאלה 3 - Alpha-Beta Pruning

כדי לחקור ביעילות רבה יותר את עץ Pruning Beta-Alpha צורו סוכן חדש שמשתמש ב שוב, האלגוריתם שלכם יהיה קצת יותר כללי מהפסאודוקוד. AlphaBetaAgent -המינימקס, ב בהתאם למספר גורמי Pruning Beta-Alpha מההרצאה, כך שחלק מהאתגר הוא להרחיב את minimizer.

- יפעל מהר כמו מינימקס עומק Beta-Alpha אתם אמורים לראות שיפור במהירות (אולי עומק 3 ב אמור לרוץ תוך כמה שניות לכל מהלך או מהר smallClassic 2 .) באופן אידיאלי, עומק 3 על יותר.

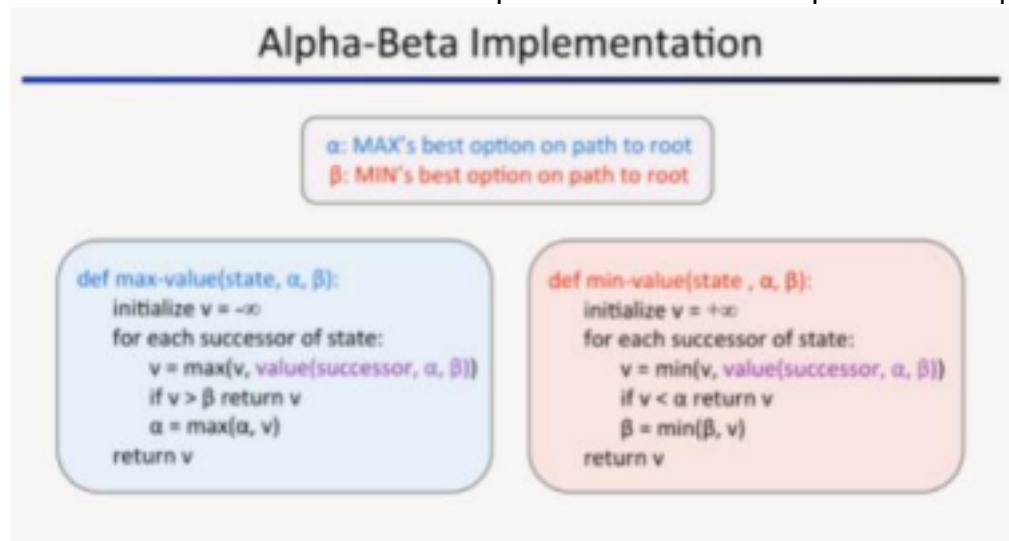
```
python pacman.py -p AlphaBetaAgent -a depth=3 -l smallClassic
```

אם כי, MinimaxAgent צריכים להיות זהים לערכי המינימקס ב AlphaBetaAgent ערכי המינימקס ב הפעולות שהוא בוחר יכולות להשתנות בגלל התנהגות שונה של שבירת קשרים. שוב, ערכי הם 9, 8, 7 ו-492 לעומקים 1, 2, 3 ו-4 minimaxClassic המינימקס של המצב ההתחלתי בפריסה בהתאמה.

שימו לב: מכיוון שאנו בודקים את הקוד כדי לקבוע אם הוא בוחר את המספר הנכון של מצבים, חשוב שתבצע גיזום אלפא-בטא מבלי לסדר ילדים מחדש. במילים אחרות, המצבים ישארו בסדר GameState.generateSuccessor-שוב, אל תקראו ל. GameState.getLegalActions שהוחזר על ידי יותר מהצורך.

שימו לב 2: לא לגזום במצבי שוויון.

קוד הפסאודו להלן מייצג את האלגוריתם שעליך ליישם עבור שאלה זו



יוביל לכך שפקמן יפסיד בחלק מההרצות. זו לא בעיה, מכיוון Pruning Beta-Alpha יישום נכון של שזו התנהגות נכונה והיא תעבור את הבדיקות.

!!בהצלחה רבה