תרגיל בית 2:

Checkers

מטרות התרגיל

- בניית שחקן עבור משחק שני שחקנים.
- התנסות בניתוח ובנייה של פונקציות יוריסטיות.
- התנסות באלגוריתם $Minmax$ ומימוש וריאציות עליו.
- ביצוע מחקר המשווה בין ביצועי שחקנים שונים.

הערות

- תאריך הגשה: 16/12/2020.
- את המטלה יש להגיש **בזוגות בלבד!** (לא יחידים, לא שלשות...)
- קראו היטב את ההסברים וההוראות במסמך זה, מטרתם לסייע לכם בהבנת הדרישות של התרגיל. שימו לב להוראות ההגשה המצורפות בסוף התרגיל.
- לתרגיל מצורף קוד חלקי הממש את המשחק. קראו היטב את ההערות הנמצאות בקוד – הם שם כדי לעזור לכם. ☺
- הִתְעַדְכְּנוּ ברשימת ה-FAQ שב classroom בתדירות גבוהה, שם אתם יכולים לשלוח את שאלותיכם. ההערות שתתפרסמנה באתר הקורס מחייבות את כלל הסטודנטים בקורס!
- הקוד שלכם ייבדק על-ידי $Unit-Tests$, לכן עליכם לעקוב בִּתשומת לב רבה אחר הוראות ההגשה המצורפות במהלך התרגיל ובסופו לפני הגשתו.
- הרצת הניסויים עשויה לקחת זמן רב ולכן מומלץ מאוד להימנע מדחיית העבודה על התרגיל לרגע האחרון. **לא תינתנה דחיות על רקע זה.**

מבוא והנחיות

במטלה זו תתכננו ותממשו שחקנים למשחק $Checkers$ (דמקה). את חוקי המשחק המחייבים לצורך התרגיל ניתן למצוא כאן: https://www.itsyourturn.com/t_helptopic2030.html. הקפידו לקרוא אותם בקפידה טרם העבודה על התרגיל.

לצורך התרגיל, נגדיר מספר חוקים החורגים מההגדרות המופיעות באתר (ממומש כבר בקוד):

1. ביצוע "קפיצה" ($capture$) – אם למשתמש קיימת אפשרות לבצע קפיצה, עליו לקחת אותה. אם קיימות מספר אפשרויות לקפיצה (למשל ע"י כלים שונים), הוא רשאי לבחור כל אחת מהפעולות. אם בסיום הקפיצה קיימת אפשרות לקפיצה נוספת בכיוון התנועה של הכלי (כלומר קדימה בלבד לשחקנים פשוטים), על השחקן לבצע את הקפיצה הנוספת. כך ימשיך לבצע כל קפיצה נוספת עד שיעצר בנקודה בה לא ניתן לבצע עוד קפיצה. למען הנוחות, נקרא למהלך כזה "קפיצה מרובה".
2. סיום המשחק ב"תיקו" – האופן היחיד בו משחק יסתיים בתיקו הוא כאשר מס' התורות מאז הקפיצה האחרונה יעלה על 50. לצורך זה כל זוג מהלכים (אחד מכל שחקן) יחשב לתור. לשחקנים לא קיימת האפשרות להציע תיקו על דעת עצמם.

השחקנים שתממשו, בהינתן מצב לוח, יחזירו את הפעולה לביצוע. בחירת הפעולה תעשה באמצעות ואריאציות שונות לאלגוריתם $Minmax$ עם גיזום $\alpha\beta$. על כן, מומלץ לחזור על שקפי ההרצאות והתרגולים בנושא "משחקים" לפני תחילת העבודה על התרגיל. בנוסף, השחקן יהיה מחויב לשתי מגבלות זמן:

1. זמן אתחול (setup): מגבלה על זמן הריצה של הבנאי ($constructor$). בזמן זה יוכל השחקן להתכונן למשחק, לאתחל מבני נתונים וכו'. לצורך התרגיל (והתחרות) נקבע זמן זה להיות 2 שניות.
2. זמן עבור k מהלכים: **מגבלה על הזמן הלוקח לשחקן לבחור את המהלך הבא**. לכל k מהלכים במשחק יש אותה מגבלת זמן, כאשר על השחקן לבחור איך לחלק את הזמן בין המהלכים. זמן שאינו מנוצל בתום k מהלכים אינו נצבר לטובת k המהלכים הבאים. בתרגיל (ובתחרות): $k = 5$.

בעוד שזמן האתחול הוא קבוע, הזמן עבור k מהלכים עשוי להשתנות בין משחקים, ולכן השחקנים שתממשו יפעלו במתכונת anytime-contract עם מדיניות ניהול זמנים לבחירתכם.

(הזמן לחלוקה בין K המהלכים מתקבל כפרמטר, וחלוקת הזמן בינהם מתבצעת באמצעות פו' - הסבר בקוד)

לפני שאתם ניגשים לתרגיל, מומלץ להתנסות מעט במשחק על-מנת לקבל הבנה טובה יותר שלו. ניתן למצוא גרסה חינמית של המשחק כאן: http://www.247checkers.com/. כמו כן, **בקוד שסופק לכם** קיים מנוע להרצת המשחק, שחקן אינטראקטיבי (המחכה לקלט מהמשתמש לשם ביצוע מהלך), וכן שחקן רנדומלי ושחקן עם היוריסטיקה פשוטה, אשר ניתן לשחק מולם.

חלק א' – הבנת המשחק (20 נק')

בשלב ראשון, נרצה לקבל הבנה בסיסית להגדרת מרחב המצבים מעליו נפעל. שימו לב כי בחלק זה (חלק א') **<u>אין צורך לכתוב קוד</u>**.

1. פתחו את הקובץ $checkers/board.py$ והסתכלו על המחלקה $GameState$. מהו המבנה של כל מצב ($state$) במרחב המצבים של המשחק?
2. במחלקה $GameState$, הפונקציה $get\_possible\_moves$ מחזירה את כל המהלכים (אופרטורים) שניתן לבצע מהמצב הנוכחי. כל מהלך מוחזר כאובייקט $GameMove$ המוגדר בקובץ $checkers/moves.py$. איזה מידע נחוץ כדי לייצג כל אופרטור? איך ניתן להבדיל מעל מבנה זה בין תנועה רגילה ($movement$) לפעולות קפיצה ($capture$)?

חלק ב' – הבנת השחקן הפשוט (12 נק')

קראו את הקוד של השחקן הפשוט המסופק כדוגמא ב $players/simple\_player$. שימו לב כי בחלק זה (חלק ב') **<u>אין צורך לכתוב קוד</u>**.

3. מהי הגישה (הנאיבית) בה נוקט השחקן לשם חלוקת הזמן בין כל k מהלכים?
4. ציינו חסרון אפשרי של גישה זו והסבירו כיצד שחקן חכם יותר יוכל להתגבר עליו.
5. מהי הפונקציה היוריסטית בה משתמש השחקן? הסבירו את המוטיבציה ליוריסטיקה זו.

חלק ג' – שיפור השחקן (24 נק')

בחלק זה של התרגיל תדרשו לממש שלושה שחקנים. על כל שחקן להיות ממומש בתיקייה נפרדת על פי שמו. מחלקת השחקן תקרא $Player$ ועליה לרשת מהמחלקה $AbstractPlayer$ (או מהמחלקה $Player$ של $simple\_player$).

6. <u>שיפור היוריסטיקה</u>:
   I. הגדירו באופן מפורש יוריסטיקה משלכם להערכת מצבי המשחק. בפרט, יש לספק נוסחה עבור כל פרמטר של המשחק הנלקח בחשבון לשם חישוב היוריסטיקה.
   II. הסבירו את המוטיבציה להגדרה זו (ולכל פרמטר המופיע בה). מדוע אתם צופים שהיא תשפר את ביצועי השחקן ביחס ליוריסטיקה של $simple\_player$?
   III. ממשו שחקן בשם $better\_h\_player$ הנעזר ביוריסטיקה שהגדרתם.

כעת נרצה לממש שיפור לאלגוריתם $Minmax - \alpha\beta$ בכדי לשפר את ביצועי השחקן שלנו. את מימוש השחקן המשופר (ללא היוריסטיקה משאלה 6) יש לשמור בשם $improved\_player$.

7. <u>ניהול זמנים</u>:
   I. הגדירו שיטה אינטליגנטית לחלוקת הזמן של השחקן בין כל k מהלכים. שיטה נאיבית ניתן, כאמור, למצוא בשחקן המסופק לכם. יש להציג נוסחה כללית במקרה הצורך, ונוסחה עבור כל פרמטר של המשחק הנלקח בחשבון לשם חישוב זה.
   II. הסבירו את המוטיבציה מאחורי השיטה שמימשתם. באילו מצבים השיטה משקיעה פחות זמן חיפוש, ובאילו יותר? מדוע לדעתכם פעולה זו תהיה משמעותית לשיפור השחקן הפשוט?
   III. ממשו שחקן עם מדיניות ניהול הזמנים שהגדרתם.

8. ממשו שחקן בשם $improved\_better\_h\_player$ המשתמש ביוריסטיקה ובשיפור הזמנים בשאלות 6-7.

חלק ד' – ניסויים, תוצאות ומסקנות (44 נק')

נרצה להשוות בין שחקני אלפא-בטא הנעזרים ביוריסטיקות השונות שראינו, עם ובלי שיפורים, תחת מגבלות זמן שונות. בפועל, ההשוואה תיעשה בין 4 שחקנים:

(1) $simple\_player$ המסופק (שחקן $baseline$)

(2) $better\_h\_player$ שמימשתם בשאלה 6

(3) $improved\_player$ שמימשתם בשאלה 7

(4) $improved\_better\_h\_player$ שמימשתם בשאלה 8

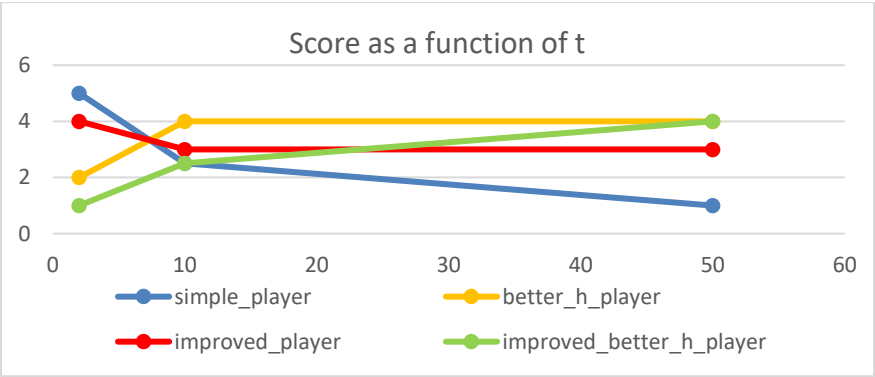
נסמן $T = \{2,10,50\}$. עבור כל זוג שחקנים (סה"כ $\binom{4}{2}$ צמדים) הריצו סדרת משחקים עבור כל מגבלת זמן $t \in T$ (בשניות). להזכירכם, t מגביל את זמן החישוב של השחקן עבור k מהלכים (כפי שהוסבר במבוא). בכל מגבלה, הריצו 2 משחקים, אחת לכל חלוקת צבעים לשחקנים. סה"כ עליכם להריץ <u>36 משחקים</u>.

לצורך ניתוח התוצאות, נשתמש בשיטת הניקוד הבאה:

- <u>הניקוד למשחק יחיד</u> - ניצחון = 1 נק', תיקו = 0.5 נק', הפסד = 0 נק'.
- <u>הניקוד הכולל של שחקן</u> יחושב כסכום הנקודות שלו מכל המשחקים בהם הוא לקח חלק.

9. הגישו קובץ בשם $experiments.csv$ שיכיל את תוצאות המשחקים שהרצתם. כל שורה בקובץ תכיל 5 עמודות: שם השחקן האדום (מבין 4 השחקנים המשתתפים בניסוי), שם השחקן השחור, מגבלת הזמן בשניות ל$k = 5$ תורות, ניקוד השחקן האדום, ניקוד השחקן השחור.
לנוחיותכם מסופקת שורה לדוגמא:
$simple\_player, improved\_player, 10,0.5,0.5$
אין לכלול כותרות בקובץ, אלא רק את תוצאות הניסויים.

10. נרצה להשוות את ביצועי השחקנים על פני מגבלות הזמן השונות:
   ו. הציגו גרף המתאר עבור כל אחת ממגבלות הזמנים את <u>הניקוד הכולל של כל שחקן</u> כפונקציה של מגבלת הזמן. הקפידו להשתמש בצבעים או בסוגי קו שונים עבור שחקנים שונים. לנוחיותכם מצורף גרף לדוגמא $(*)$.



II. עבור גרף זה, צרפו גם את טבלת הנתונים שיצרו אותו. למשל(*):

	$t = 2$	$t = 10$	$t = 50$
$simple\_player$	5	2.5	1
$better\_h\_player$	2	4	4
$improved\_player$	4	3	3
$improved\_better\_h\_player$	1	2.5	4

III. השוו את ביצועי השחקנים עם היוריסטיקה הפשוטה (שחקנים (1) ו(3)) לאלו עם היוריסטיקה שהצעתם בשאלה 6 (שחקנים (2) ו(4)) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה בניקוד הכולל של כל אחד מהשחקנים. באילו מקרים היוריסטיקה שלכם מתגברת על היוריסטיקה הפשוטה? הציעו הסבר לממצאים הנ"ל.

IV. השוו את ביצועי השחקנים בעלי השיפור משאלה 7 (שחקנים (3) ו(4)) לאלו ללא שיפור זה (שחקנים (1) ו(2)) ביחס למגבלת הזמן ע"י ניתוח של מגמות השיפור והדעיכה של כל אחד מהשחקנים. באילו מקרים השיפור שלכם משפר את ביצועי השחקן? הציעו הסבר לממצאים הנ"ל.

(*) שימו לב כי הגרפים והטבלאות שהצגנו כאן נוצרו על סמך ערכים שרירותיים שקבענו לצורך **המחשה בלבד**. לא מובטח שהתוצאות שתקבלו בהרצת הניסויים שלכם תהיינה דומות לערכים המוצגים בגרף ובטבלה שלעיל.

בהצלחה!

הוראות הגשה

- הגשת התרגיל תתבצע **אלקטרונית בלבד** דרך אתר הקורס.
- כהכנה להגשה, אתם מתבקשים ליצור בתוך התיקייה $players$ שבקוד המסופק **תיקיה יחידה** בשם $AI2\_\langle id1\rangle\_\langle id2\rangle$ (ללא הסוגריים המשולשים), שבתוכה הקבצים הבאים:
  - קובץ בשם $readme.txt$ בפורמט הבא:

name1 id1

name2 id2

  - קובץ ריק בשם $\_\_init\_\_.py$
    תיקיה עבור השחקנים שקיבלתם והשחקנים שמימשתם:
    $better\_h\_player, improved\_player, improved\_better\_h\_player, simple\_player$
    $random\_player, interactive$
    על כל אחת מהתיקיות הללו להכיל קובץ $\_\_init\_\_.py$ עם קוד השחקן שלכם.
    מחלקת השחקן צריכה להיקרא $Player$ ולרשת מהמחלקה $AbstractPlayer$ (או מהמחלקה $Player$ של $simple\_player$) שבקוד המסופק.
  - כל קובץ עזר שכתבתם, אשר קוד השחקנים משתמש בו.
  - קובץ בשם $requirements.txt$ שיכיל כל חבילה חיצונית שאינה מותקנת כחלק מ-$Anaconda\ Python$. על אחריותכם לוודא כי ניתן להתקין כל חבילה כנ"ל באמצעות הרצת השורה: $pip\ install - r\ requirements.txt$
  - קובץ בשם $AI\_HW2.pdf$, המכיל את התשובות לחלק היבש והערות מיוחדות הנוגעות לקוד במידה ויש צורך בכך.

- על-מנת לוודא שמבנה הקבצים הנ"ל תקין, בדקו שאתם מסוגלים להריץ את הפקודה הבאה:

$$python\ run\_game.py\ 2\ 3\ 5\ y\ AI3\_\langle id1\rangle\_\langle id2\rangle.improved\_player\ simple\_player$$

פקודה זו תריץ משחק בין השחקן המשופר שלכם (בתור השחקן האדום) לבין השחקן הפשוט.

- את התיקייה $AI2\_\langle id1\rangle\_\langle id2\rangle$ יש לקבץ לארכיון בשם $AI2\_\langle id1\rangle\_\langle id2\rangle.zip$, **שאותו (ואותו בלבד) עליכם להגיש.;;;;1**
- אין להעתיק את הקבצים המסופקים לכם אל תוך תיקיית ההגשה. הניחו כי קבצים אלו יהיו זמינים בעת בדיקת התרגיל.
- שימו לב שכל הפנייה למיקום קובץ/תיקייה כלשהם בקוד תהיה רלטיבית ($relative\ path$) ולא אבסולוטית, כך שהקוד יעבוד כפי שהוא על כל מחשב בכל מיקום שנבחר לתיקיית הפרויקט. הקפידו לבדוק זאת לפני ההגשה!
- הקפידו על קוד **<u>ברור, קריא ומתועד!</u>** עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם. בפרט, אם התשמשתם בקוד שנמצא ברשת וביצעתם בו שינויים, עליכם לתעד זאת.
- אתם רשאים לעשות שימוש בכל קוד שתמצאו ברשת, אך כל קוד חיצוני מצריך הצהרה מפורשת על המקור שלו בקובץ $AI\_HW2.pdf$.