

## תרגיל בית 2: משחקים - Lasca

### מטרות התרגיל

- הבנת הדינמיקה של משחקי שני שחקנים על בסיס משחק ספציפי.
- התנסות בהגדרת פונקציות היוריסטיות.
- התנסות באלגוריתם מינימקס אלפא-בטא ובמימוש וריאציות עליו.
- ביצוע מחקר המשווה בין ביצועי שחקנים שונים.

### הערות

- תאריך הגשה: 01/01/15.
- את המטלה יש להגיש בזוגות בלבד.
- שאלות בנוגע לתרגיל יש לשלוח למיטל: [maytal@cs.technion.ac.il](mailto:maytal@cs.technion.ac.il).
- אנא עיינו ברשימת ה-FAQ המתעדכנת באתר לפני פנייה במייל.



### מבוא

במטלה זו תתכננו ותממשו שחקן למשחק Lasca. השחקן, בהינתן מצב לוח, יספק את הפעולה לביצוע. השחקן יהיה מחויב לשתי מגבלות זמן:

1. זמן אתחול (setup): הזמן הנדרש כדי להתכונן למשחק, לאתחל מבני נתונים וכו'. לצורך התרגיל והתחרות נקבע זמן זה להיות 2 שניות.
2. זמן עבור  $k$  מהלכים: לכל  $k$  מהלכים במשחק יש אותה מגבלת זמן, כאשר זמן שאינו מנוצל בתום  $k$  מהלכים אינו נצבר לטובת  $k$  המהלכים הבאים. בתרגיל ובתחרות:  $k=5$ .

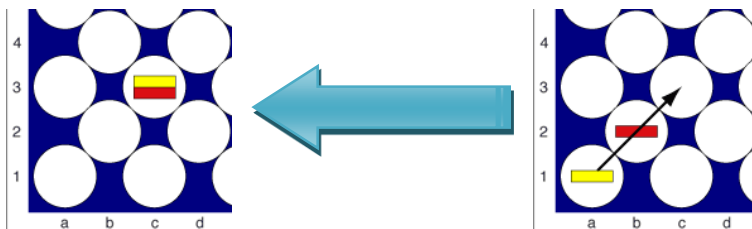
בעוד שזמן ה-setup הוא קבוע, הזמן עבור  $k$  מהלכים עשוי להשתנות בין משחקים, ולכן יהיה עליכם לממש את השחקן במתכונת anytime-contract עם מדיניות ניהול זמנים לבחירתכם.

בנוסף, נגביל את מספר המהלכים הכולל לשחקן במשחק ל-100, על מנת להימנע מכניסה ללולאה אינסופית של מהלכים חוזרים. חריגה ממגבלה זו תגרור את הפסקת המשחק בתוצאה תיקו.

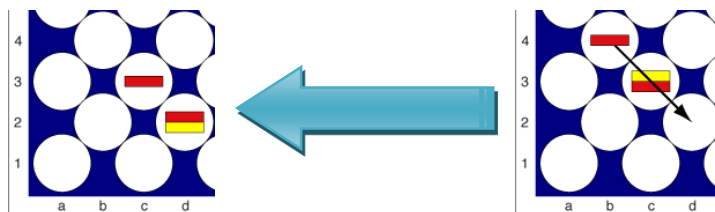
## תיאור המשחק

Lasca הינו משחק לוח המבוסס על דמקה. לוח המשחק הוא בן  $7 \times 7$  משבצות, כאשר התזוזה של הכלים מתבצעת רק באלכסון, משבצת אחת בכל פעם. בתחילת המשחק, כל הכלים הם בעלי מעמד של **חילים**, והם רשאים לנוע קדימה בלבד (אל עבר מרכז הלוח). השחקן הלבן תמיד פותח את המשחק.

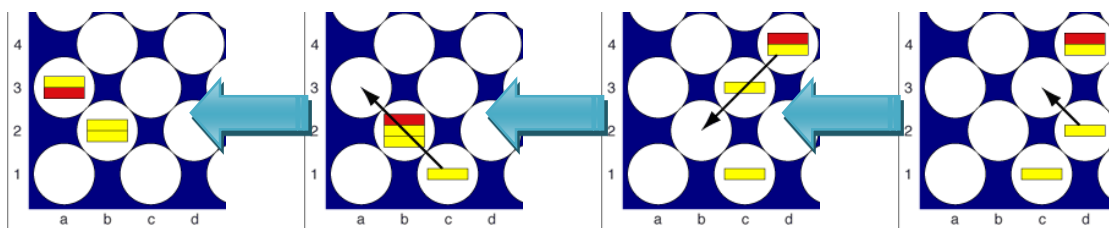
ניתן ללכוד כלי של היריב אם מיד מאחוריו יש משבצת ריקה, זאת ע"י דילוג מעליו אל המשבצת הריקה. הכלי המותקף אינו מוסר מהלוח, אלא ממוקם מתחת לכלי התוקף והופך לשבוי שלו. ערימת הכלים שנוצרת כתוצאה ממהלך הלכידה מכונה **עמודה**, כאשר הכלי העליון הוא המנהיג שלה:



כל הכלים בעמודה זזים יחד, אך לא ניתן לקחת יותר משבוי אחד מכל משבצת. כלומר, אם ליריב יש עמודה שניתן לדלג מעליה, רק המנהיג של העמודה (הכלי העליון ביותר) ילקח כשבוי:



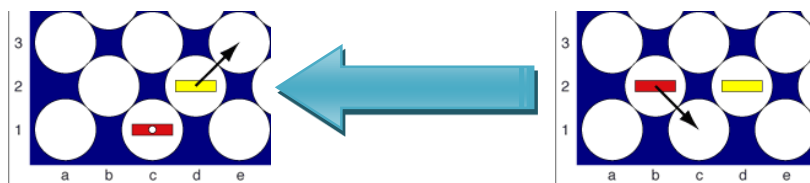
שחקן חייב לבצע לכידה של כלי אם באפשרותו לעשות זאת:



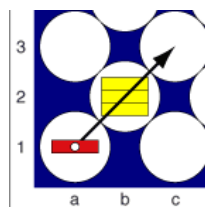
אם מיד לאחר לכידה של כלי מתאפשר לכלי הלכוד לבצע לכידות נוספות, הוא מחויב לבצע גם אותן.

כאשר עמודה מגיעה לצד הנגדי של הלוח (הקו האחרון של האויב), המנהיג של העמודה מקודם לדרגת קצין (מסומן בנקודה באיור). כלי בודד מקודם באופן דומה. קצין, או עמודה שמנהיגה הוא קצין, יכולים לנוע קדימה או אחורה (באלכסון).

בדוגמה שלמטה, הקידום של הכלי לדרגת קצין מתרחש רק עם תום המהלך, ולכן הוא אינו רשאי להמשיך וללכוד את הכלי הלבן באותו תור:



אם במהלך שרשרת פעולות לכידה הנעשות באותו מהלך, עמודה שמנהיגה הוא חייל פשוט מגיע לקו האחרון של האויב, מנהיג העמודה מקודם לדרגת קצין ותורו מסתיים (הוא אינו רשאי להמשיך ללכוד כלים). לעומת זאת, אם המנהיג היה קצין מלכתחילה, הוא רשאי (ומחויב) ללכוד כלים נוספים באותו מהלך אם הדבר מתאפשר.



שחקן אינו רשאי לתקוף את אותו כלי פעמיים במהלך בודד. לכן בדוגמה הבאה, הקצין רשאי לדלג מעל העמודה פעם אחת בלבד, ולא 3 פעמים:

קצין שנלקח בשבי, שומר על דרגתו למקרה שישוחרר בהמשך המשחק.

שחקן שכל כליו נלקחו בשבי, או שלא נותרו לו מהלכים חוקיים, מפסיד במשחק.

## טיפים והנחיות

לפני שאתם ניגשים לתרגיל, מומלץ להתנסות מעט במשחק על-מנת לקבל הבנה טובה יותר שלו. מספר אופציות להתנסות:

- משחק חינמי להורדה לאנדרואיד:  
<https://play.google.com/store/apps/details?id=com.android.laska>
- גרסת אונליין של המשחק: [http://www.pjb.com.au/laska/play\\_laska.html](http://www.pjb.com.au/laska/play_laska.html)
- בקוד שסופק לכם קיים מנוע להרצת המשחק, שחקן אינטראקטיבי (המחכה לקלט מהמשתמש לשם ביצוע מהלך), וכן שחקן רנדומלי ושחקן עם היוריסטיקה פשוטה, אשר ניתן לנסות לשחק מולם.

לנוחיותכם מספר אופציות להורדה של הקוד המסופק:

- כקובץ zip דרך האתר (Download ZIP): [https://github.com/TechnionAI/Win14\\_15\\_HW2](https://github.com/TechnionAI/Win14_15_HW2)
- על מנת ליצור עותק מקומי (clone) של הקוד, הריצו בסביבת Unix את הפקודה:  
`git clone git@github.com:TechnionAI/Win14_15_HW2.git`
- מספיק להריץ פקודה זו פעם אחת. על מנת למשוך שינויים עתידיים בקוד אם יהיו, פשוט הריצו:  
`git pull`
- למי שפחות נח לו לעבוד עם ה-shell, קיימים גם שלל כלים גרפיים לעבודה עם git בסביבת חלונות. פשוט חפשו "git for windows".

**חלק א' – הבנת שחקן הבסיס (15 נק')**

קראו את הקוד של שחקן ה-baseline המסופק כדוגמא ב-ai\_game\_hw2/players/simple\_player.

1. מהי הגישה (הנאיבית) בה נוקט השחקן לשם חלוקת הזמן בין המהלכים?
2. ציינו חיסרון אפשרי של גישה זו והסבירו.
3. מהי הפונקציה ההיוריסטית בה משתמש השחקן? הסבירו את המוטיבציה להיוריסטיקה זו.

**חלק ב' – שיפור השחקן (40 נק')**

4. הגדירו באופן פורמלי היוריסטיקה משלכם להערכת מצבי המשחק. יש להגדיר מפורשות כל פרמטר של המשחק, אשר נלקח בחשבון לשם הרכבת ההיוריסטיקה. הסבירו את המוטיבציה להגדרה זו ומדוע אתם צופים שהיא תשפר את ביצועי השחקן ביחס להיוריסטיקה של השחקן שסופק לכם.
  5. בחרו וממשו אחד מהשיפורים הבאים לאלגוריתם מינימקס אלפא-בטא, שלדעתכם יהיה אפקטיבי בשיפור ביצועי השחקן:
    - סידור בנים – הציעו דרך לקביעת הסדר בו יפותחו הבנים של כל צומת. הסבירו את הרציונל מאחורי הדרך שהצעתם.
    - העמקה סלקטיבית עד רגיעה - הציעו קריטריון למצב "לא שקט" (מצב הנראה מבטיח בטווח הקצר, אך הבחירה בו עשויה להתגלות כמזיקה תוך מספר מהלכים). תארו את הרציונל מאחורי הקריטריון שהגדרתם. תזכורת: על מנת לקבל הערכה טובה יותר של מצבים "לא שקטים", נרצה להמשיך לפתח אותם עד ל"רגיעה" (גם אם הגענו כבר למגבלת עומק החיפוש).
    - ניהול זמן – הגדירו שיטה אינטליגנטית לחלוקת הזמן של השחקן בין כל k מהלכים. שיטה נאיבית ניתן, כאמור, למצוא בשחקן המסופק לכם.
- ציינו מפורשות את בחירתכם והסבירו מדוע לדעתכם שינוי זה יהיה משמעותי בשיפור השחקן.

**חלק ג' – ניסוי וניתוח תוצאות (45 נק')**

מטרת הניסוי: להשוות בין שחקני אלפא-בטא עם היוריסטיקות שונות, עם ובלי שיפורים, תחת מגבלות זמן שונות. לצורך הניסוי, נגדיר ארבעה שחקנים, היורשים מהמחלקה AbstractPlayer שבקוד המסופק:

- שחקן ה-baseline (מסופק).
- שחקן אלפא-בטא המשתמש בהיוריסטיקה שהגדרתם בשאלה 4.
- שחקן ה-baseline בתוספת השיפור שהגדרתם בשאלה 5.
- שחקן אלפא-בטא המשתמש בהיוריסטיקה שהגדרתם בתוספת השיפור שהצעתם.

עבור כל זוג שחקנים  $\binom{4}{2}$  צמדים כפול 2 אפשרויות לבחירת צבעי השחקנים), הריצו סדרת משחקים

עם מגבלות הזמנים הבאות: 2.5 שניות, 10 שניות, 40 שניות (כל מגבלת זמן היא עבור 5 מהלכים). סה"כ עליכם להריץ 36 משחקים.

6. בסעיף זה נסתמך על שיטת הניקוד הבאה:

- ניצחון = 1 נק'
- תיקו = 0.5 נק'
- הפסד = 0 נק'

סכמו בטבלה את הניקוד הכולל של כל שחקן כפונקציה של מגבלת הזמן (שימו לב שהניקוד המקסימלי לשחקן תחת מגבלת זמנים ספציפית הוא 6). הציגו את התוצאות בגרף בודד. הקפידו להשתמש בצבעים או בסוגי קו שונים עבור שחקנים שונים.

7. כיצד משתנה, אם בכלל, דירוג השחקנים כאשר בוחנים מגבלות זמן שונות? במקרים בהם אין שינוי בדירוג היחסי בין שני שחקנים, האם פער הנקודות ביניהם הצטמצם או גדל? הציגו הסבר לממצאים. הקפידו להתייחס ולהציע הסבר לכל תוצאה הנוגדת את האינטואיציה.
8. בהסתמך על הממצאים, הציגו דרך אחת לפחות באמצעותה ניתן לנסות לשפר את השחקן. הסבירו על אילו ממצאים הסתמכתם בהצעתכם וכיצד.

## חלק ד' – התחרות

עליכם לממש שחקן יחיד עבור השתתפות בתחרות. ניתן להשתמש באחד השחקנים שמימשתם בסעיפים הקודמים או לחילופין לממש שחקן חדש כרצונכם, אך ההשתתפות בתחרות היא חובה. למטלה זו אין משקל בניקוד תרגיל בית 2 עם סייג אחד: **במידה והשחקן שתגישו לא ירוץ, תיתכן הורדת נקודות על התרגיל.**

בתחרות שנקיים בין השחקנים שתגישו, יאכפו שתי מגבלות זמנים: מגבלת זמן גלובלית של 2 שניות לשחקן עבור שלב האתחול, וכן מגבלת זמן גלובלית של מספר שניות לשחקן עבור כל 5 מהלכים. שימו לב שהדבר מאפשר לכם חופש פעולה בהגדרת הזמן שברצונכם להקצות לחישוב כל אחד מהמהלכים. הסוכנים ייבחנו תחת 3 מגבלות הזמנים שהוגדרו בשלב הניסוי. זמן החישוב הכולל של כל 5 מהלכים צריך להיות נמוך מהמגבלה המוגדרת. שחקן שיחרוג מהזמן המוקצה, יפסיד טכנית במשחק הספציפי בו קרתה החריגה.

- מחלקת השחקן צריכה להיקרא `Player` ולרשת מהמחלקה `AbstractPlayer`.
  - אתם רשאים לבחור עבור השחקן שלכם שם יצירתי כרצונכם ע"י דריסת הפונקציה `__repr__` ב-`Player` באופן הבא:
- ```
def __repr__(self):
    return '{} {}'.format(abstract.AbstractPlayer.__repr__(self),
                           'Senorita_Analphabeta')
```
- חל איסור להשתמש ברשת האינטרנט או ברשת מקומית כלשהי.
  - על השחקן שלכם לעבוד כחוט יחיד.
  - צריכת הזיכרון של השחקן שלכם מוגבלת ל-512 MB.
  - כל שחקן שינסה לרמות או לשבש את קוד היריב או המערכת ייפסל.
  - ניתן לבצע פעולות אתחול ו/או עיבוד מקדים בבנאי (`__init__`) של השחקן.

- התחרות תכלול שלושה מקצים עם אותן מגבלות זמן שהוגדרו בחלק ג' של התרגיל.
- סטודנטים אשר יממשו שחקנים שיצטיינו בתחרות, יזכו בבונוס מכובד לציון הסופי של הקורס (ובתהילת עולם):



## הוראות הגשה

- הגשת התרגיל תתבצע אלקטרונית בלבד.
- כהכנה להגשה, אתם מתבקשים ליצור בתוך התיקייה `players` שבקוד המסופק **תיקייה יחידה** בשם `AI2_<id1>_<id2>` (ללא הסוגריים המשולשים), שבתוכה הקבצים הבאים:
  - קובץ `__init__.py` בשם `__init__.py`.
  - 4 תיקיות ששמן `competition_player`, `player3`, `player2`, `player1` עבור 3 השחקנים שמימשתם עבור חלק ג' של התרגיל והשחקן אותו אתם מגישים לתחרות. על כל אחת מהתיקיות הללו להכיל קובץ `__init__.py` עם קוד השחקן שלכם (או `import` לקוד). מחלקת השחקן צריכה להיקרא `Player` ולרשת מהמחלקה `AbstractPlayer` שבקוד המסופק.
  - כל קובץ עזר שכתבתם, אשר קוד השחקנים משתמש בו.
  - כל חבילה חיצונית בה השתמשתם (שלא אנחנו סיפקנו לכם), **אשר לא ניתן להתקין** באמצעות הפקודה `pip install`, זאת על מנת שיהיה אפשר להריץ את הקוד שלכם על כל מחשב.
  - קובץ `libs.txt`, המכיל את שמות כל החבילות בהן השתמשתם ואשר **ניתן להתקין** באמצעות הפקודה `pip install`. הקובץ יכול להיות ריק. הקובץ צריך להיות בפורמט הבא:
 

```
name1 id1 email1
name2 id2 email2
```
- קובץ בשם `AI_HW2.PDF`, המכיל את התשובות לחלק היבש והערות מיוחדות הנוגעות לקוד במידה ויש צורך בכך.
- קובץ בשם `README.txt` בפורמט הבא:

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

```
python run_game.py 3 3 3 50 t AI2_123_456.competition_player simple_player
```
- פקודה זו אמורה להריץ משחק בין השחקן שלכם (כשחקן הלבן) לבין שחקן ה-baseline. את התיקייה <id1>\_<id2> AI2 יש לקבץ לארכיון בשם <id1>\_<id2> AI2, **שאותו (ואותו בלבד) עליכם להגיש.**
- 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 ברור, קריא ומתועד ברמה סבירה. עליכם לתעד כל חלק שאינו טריוויאלי בקוד שלכם.
- אתם רשאים לעשות שימוש בכל קוד שתמצאו ברשת, אך כל קוד חיצוני מצריך הצהרה מפורשת על המקור שלו בקובץ .AI\_HW2.PDF

**בהצלחה!**