

## תוכנה 1 – אביב תשע"ג

### תרגיל מספר 12

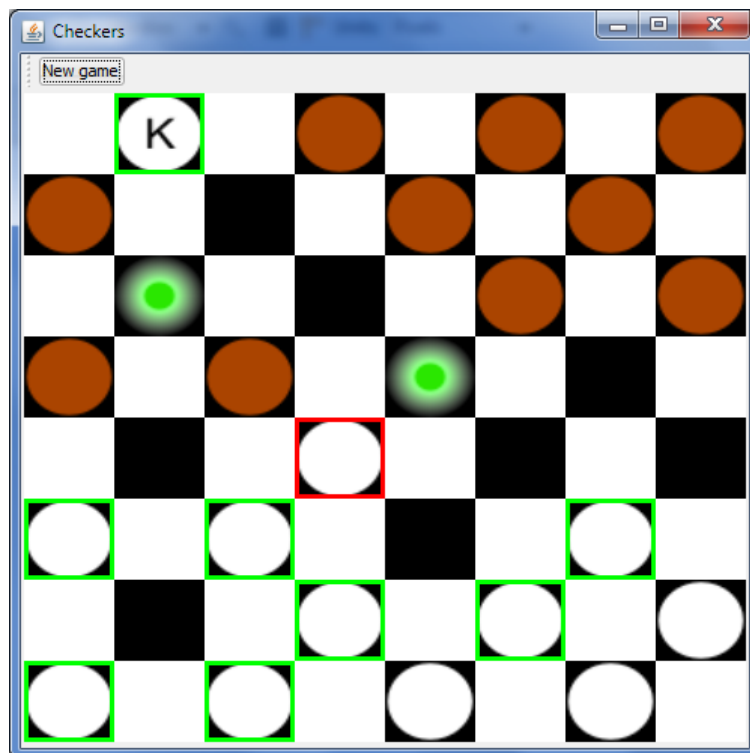
#### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer\_hw12.zip). קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.
  - ג. קובץ טקסט אחד עם העתק של כל קבצי ה-java.

#### דמקה

בתרגיל זה נכתוב משחק דמקה פשוט. התרגיל יכלול מימוש של קוד הקשור הן ללוגיקת המשחק והן לממשק המשתמש.

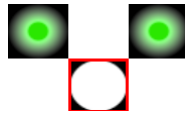


בכל הדוגמאות הגרפיות שנציג נשתמש בסימונים הבאים (מודגמים בציור למעלה):

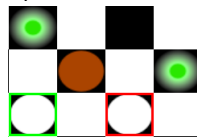
- משבצות של כלים שיכולים לבצע מהלך יסומנו במסגרת ירוקה.
- המשבצת של הכלי שנבחר לבצע מהלך תסומן באדום והמהלכים האפשריים ייוצגו ע"י עיגולים ירוקים במשבצות היעד.
- חייל יסומן ע"י עיגול ומלך יסומן ע"י האות K בתוך העיגול.

חוקי המשחק:

1. ישנם שני שחקנים, שחקן אחד משתמש בכלים הלבנים (השחקן הלבן) והשני בכלים הכהים.
2. לכל שחקן מותר להזיז רק את הכלים שלו.
3. השחקן הלבן מתחיל את המשחק.
4. כל שחקן מתחיל עם 12 כלים הממוקמים בשלוש השורות הראשונות של הלוח, על גבי המשבצות השחורות.
5. מותר לזוז רק על גבי המשבצות השחורות על פי החוקים הבאים:
  - א. **תזוזה.** לחייל מותר לזוז באלכסון למשבצת סמוכה הנמצאת בשורה הבאה אם היא פנויה.

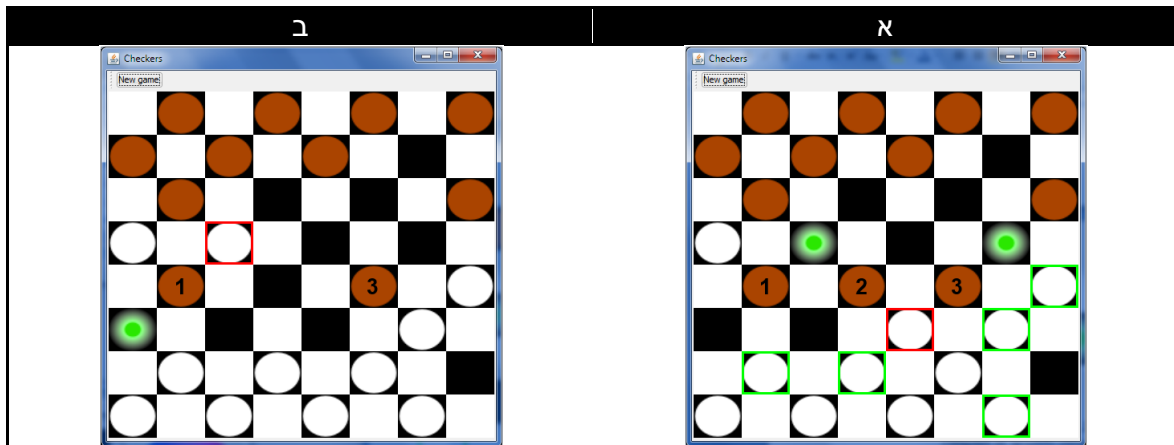


- ב. **שבי (אכילה).** אם המשבצת הסמוכה באלכסון מכילה כלי של השחקן יריב, ניתן לקפוץ מעליו למשבצת פנויה הסמוכה לה עם כוון התנועה, ולשבות (לאכול) את הכלי (להוציאו מן המשחק).



- i. כל עוד ניתן לשבות כלים נוספים בכל כיוון שהוא (קדימה או אחורה) – חובה לעשות כן.
- ii. ניתן לבחור איזה חייל לשבות אם יש יותר מאפשרות אחת.

בדוגמא, בתמונה א', החייל הלבן הנמצא במשבצת המסומנת באדום יכול לשבות את חייל מספר 3. במקרה זה התור עובר לשחקן הכהה. לחילופין, אם החייל שובה את חייל מספר 2 (תמונה ב'), הוא מחוייב להמשיך ולשבות את חייל מספר 1.

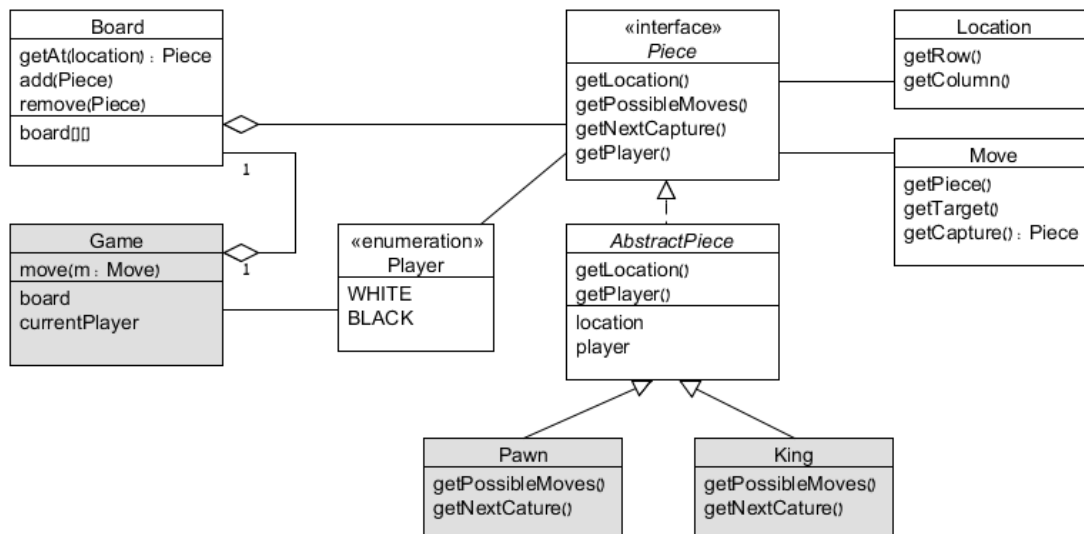


6. חייל שמגיע לשורה הרחוקה ביותר בלוח הופך למלך.
7. למלך מותר לזוז עפ"י החוקים הבאים:
  - א. מותר לזוז באלכסון, בכל כוון שהוא בתנאי שלא קופצים מעל כלים אחרים.
  - ב. אם משבצת היעד מכילה כלי של השחקן היריב ולאחריו (עם כוון התנועה) משבצת פנויה, ניתן לשבות את החייל עפ"י החוקים שפורטו עבור חייל. בפרט:
    - i. כאשר שובים כלי חייבים לקפוץ למשבצת הסמוכה לו עם כוון התנועה.
    - ii. חייבים להמשיך לשבות כלים נוספים אם ניתן לעשות כן.
8. המשחק מסתיים כאשר אחד השחקנים לא יכול לבצע לבצע מהלך, כלומר:
  - i. כאשר כל הכלים שלו נשבו
  - ii. או, כאשר נותרו כלים על הלוח אך הם אינם יכולים לזוז.

## שימו לב:

1. רוב הקוד הנדרש למימוש המשחק כבר נתון לכם. עליכם לממש רק את החלקים שיוגדרו בהמשך. יש לבצע שינויים רק במקומות המוגרים ע"י התגית TODO ולא במקומות אחרים בקוד, אלא אם צוין אחרת בסעיף הרלוונטי.
2. מכיוון שכך, בתרגיל זה אתם נדרשים לקרוא ולהבין גם קוד קיים. אנא קראו היטב את התיעוד (במסמך ובקוד).

זוהי דיאגרמת מחלקות המממשות את מבנה הנתונים ואת הלוגיקה של משחק הדמקה (אך לא את ממשק המשתמש). בחלקים א' ו-ב' של התרגיל אתם נדרשים לשנות את המחלקות Pawn, King ו-Game, המופיעות באפור.



## הסבר כללי על המחלקות

- המנשק Piece מתאר כלי במשחק. כלי יכול להיות מסוג חייל (Pawn) או מלך (King). המחלקה AbstractPiece מספקת פונקציונאליות בסיסית למימוש כלים. המחלקות הקונקרטיות Pawn ו-King מממשות את לוגיקת התנועה המתאימה לכל סוג של כלי.
- מחלקת העזר Location מתארת מיקום על הלוח (שורה ועמודה).
- מחלקת העזר Move מתארת מהלך וכוללת נתונים המציינים את: הכלי שמבצע את התנועה, משבצת המטרה אליה הוא זז וכן, אם רלוונטי, את הכלי שנשבה בעקבות התנועה.
- המחלקה Board משמשת כמבנה נתונים מטריציוני.
- המחלקה Game מנהלת את המשחק בצורה הבאה:
  - המתודה getAvailableMoves מחזירה את אוסף הכלים ואת המהלכים האפשריים לכל כלי. כדי לקבוע אילו מהלכים אפשריים, קוראים למתודה getPossibleMoves() של כל כלי.
  - המתודה move מבצעת מהלך ומחזירה את אוסף הכלים והמהלכים אותם הם יכולים לבצע לאחר ביצוע המהלך. אם המהלך האחרון גרם לשבי של כלי יריב, קוראים למתודה getNextCapture של הכלי שביצע את התנועה כדי לברר האם ניתן לשבות כלים נוספים. השחקן הנוכחי ממשיך לבצע מהלכי שבי כל עוד ניתן לעשות כן. אחרת, התור עובר לשחקן השני.

### חלק א' (30 נקודות) – שיהיה בתנועה

ממשו את המתודות getNextCapture ו-getPossibleMoves של המחלקות King ו-Pawn לפי חוקי המשחק. מותר לשנות את המתודות האלו כרצונכם.

מותר להוסיף מתודות ושדות פרטיים חדשים למחלקות אלו.

### חלק ב' (20 נקודות) – מישהו שומע אותי

המתודה move של המחלקה Game משנה את לוח המשחק. נרצה לדווח אילו שינויים בוצעו (לדוגמה כדי לעדכן את ממשק המשתמש) עבור כל מהלך. האירועים עליהם נדווח במשחק הם:

- תזוזה של חייל (PieceMoved)
- שבי של חייל (PieceCaptured)
- קידום של חייל רגלי למלך (PiecePromoted)

הערה: עבור מהלך בודד ייתכנו מספר שינויים. לדוגמה: במהלך בו חייל שובה חייל יריב ונוחת בשורה האחרונה מייצרים את האירועים: (1) שבי של חייל ו-(2) קידום של חייל למלך.

לכל אחד מהאירועים הנ"ל הוגדרו מנשק של מאזין (Listener) ומחלקה עם פרטי האירוע (Event). כאשר קורה אירוע, פרטיו מוחזקים במחלקה זו ומועברים למאזינים.

1. עליכם לממש את המתודות addXXXListener, removeXXXListener במחלקה Game כך שיתחזקו את רשימת המאזינים לאירועים השונים.

2. עליכם לשנות את המתודה move של המחלקה Game כך שתדווח על האירועים השונים כאשר הם קורים.

מותר להוסיף מתודות ושדות פרטיים חדשים למחלקה Game.

מותר להוסיף שורות קוד למתודה move אך לא לשנות או למחוק שורות קוד קיימות.

### חלק ג' (50 נקודות) – ממשק משתמש (UI)

בחלק זה נשנה שתי מחלקות הנמצאות בחבילה ui כדי להפעיל את האפליקציה הגרפית.

#### המחלקה BoardPanel

המחלקה BoardPanel היא רכיב UI האחראי על ציור הלוח ועל קליטת קלט מהמשתמש (לחיצות עכבר על הלוח). ציור הלוח נעשה באמצעות מופעים של המחלקה Square, שגם היא רכיב UI.

כאשר המשתמש לוחץ עם העכבר על ריבוע בלוח, על המחלקה BoardPanel לדווח על אירוע מתאים של בחירת מיקום, כלומר שורה ועמודה נתונים בלוח המשחק. בצורה זו, מאזינים למחלקה BoardPanel יכולים לטפל באובייקטים של המשחק (Location) במקום בקואורדינטות של עכבר.

כדי לענות על הדרישה, יש לממש מנגנון הודעות עבור אירוע שיקרא LocationSelected (המנשק LocationSelectedListener והמחלקה LocationSelectedEvent כבר הוגדרו) וכן לייצר דווח מתאים כאשר המשתמש לוחץ על העכבר. לשם כך:

1. השלימו את המתודות addLocationSelectedListener ו-removeLocationSelectedListener.
2. השלימו את המימוש של המתודה mouseReleased המסומנת בהערת TODO.

מותר להוסיף מתודות ושדות פרטיים חדשים למחלקה BoardPanel.

## המחלקה Application

המחלקה Application דואגת לעדכון התצוגה בהתאם לבחירות של המשתמש. ממשק המשתמש עובד בצורה הבאה:

- א. המשבצות בהן מצויים כלים שיכולים להתקדם מסומנות בריבוע ירוק. המתודה `highlightPieces()` של המחלקה `BoardPanel` דואגת לסמן את המשבצות המתאימות עבור אוסף החיילים המסופק כארגומנט. אם ישנו כלי יחיד שיכול לנוע, הוא נבחר (עפ"י המפורט בסעיף ב') בצורה אוטומטית.
- ב. המשתמש בוחר כלי על ידי לחיצה עם העכבר על המשבצת שלו. אם המשתמש לוחץ על משבצת שבה יש כלי שאינו יכול לבצע מהלך – מתעלמים מן הלחיצה.
  - a. משבצת של כלי שנבחר על ידי המשתמש תסומן בריבוע אדום ע"י שימוש במתודה `highlightSelected()` של המחלקה `BoardPanel`.
  - b. המשבצות אליהן יכול להגיע כלי שנבחר יסומנו בעיגול ירוק ע"י שימוש במתודה `highlightMoves()` של המחלקה `BoardPanel`.
  - ג. המשתמש בוחר מהלך על ידי לחיצה על משבצת היעד של המהלך.
    - a. אם המשתמש לחץ על משבצת חוקית – מבצעים את המהלך.
    - b. אם המשתמש לחץ על משבצת בה מצוי כלי אחר שיכול לזוז, משנים את הבחירה לכלי החדש שנבחר ע"י המשתמש.
    - c. אם המשתמש לחץ על משבצת אחרת, מבטלים את הבחירה של המשתמש.
    - ד. אם לאחר ביצוע המהלך אין מהלכים חוקיים נוספים (המשחק הסתיים)
      - a. מציגים הודעה על השחקן המנצח בעזרת המתודה הסטטית `showMessageDialog` של המחלקה `JOptionPane`.
      - b. מנקים את הסימונים על הלוח (סימוני בחירה ומהלכים). ניקוי הסימונים מתבצע ע"י קריאה למתודות המסומנות במחלקה `Board` עם פרמטר `null`.

הערה: ניתן להוריד מאתר הקורס סרט המדגים את פעולת ה-UI.

יש לממש את המתודה `locationSelected` הנקראת בתגובה לאירוע לחיצת עכבר (האירוע שמימשנו עבור המחלקה `BoardPanel`) כך שממשק המשתמש יעבוד לפי התיאור. כלומר, יש לחכות לקלט (לחיצה) ואז לקרוא למתודות המתאימות ב-`BoardPanel` להצגת חווי מתאים או ל-`Game` כדי לבצע מהלך.

מותר להוסיף מתודות ושדות פרטיים חדשים למחלקה `Application`.

# ב ה צ ל ח ה !