

עבודה 3 - עקרונות תכנות מונחה עצמים 2018

תאריך פרסום: 29.04.2018 תאריך הגשה: 16.05.2018

מתרגל אחראי: מתן קינצלינגר

1. הסבר כללי:

במשימה זו תצטרכו לממש גרסה פשוטה של המשחק Sliding Puzzle בעזרת הממשק משתמש גרפי של (java GUI SWING). התוכנית נבדקת על פי נכונות, ציות לעקרונות תכנות מונחה עצמים, שימושיות משתמש ואטראקטיביות. אתם יכולים להוסיף תוספות משלכם כל עוד עמדתם בכל הדרישות שיוגדרו בתרגיל. דרישות נוספות יכולות להעלות לכם את הציון כמפורט בסעיף הבנוס (ראו: סעיף 3). סך הציון בתרגיל יכול להגיע ל 115, כאשר 15 הם הנקודות של הבנוס. עליכם להתחיל אתכם המטלה בקריאת מסמך זה, ניתוח של הבעיה ובבנייה של תרשים מחלקות (UML) המתאים, על פי הדרישות. אם אתם לא מכירים את המשחק Sliding Puzzle, ניתן לקרוא עליו כאן, להוריד אותו מה-Google Play כאן או לשחק בו אונליין.

שימו לב!

- מומלץ מאוד (!!!) לקרוא את כל העבודה לפני תחילת המימוש.
- מנסיון עבר, ממליצים לכם לא לדלג על קריאת החלק של הוראות ההגשה - הגשה לא לפי המבנה הנכון תגרור הורדה משמעותית של נקודות.
- לכל אורך פעולתה, אסור על התוכנית לקרוס! במידה ויש מקומות בהן יש אפשרות לקריסה (כגון קריאה מקובץ וכו') התוכנית צריכה לטפל בשגיאה ולהציג הודעה מתאימה למשתמש.

2. פירוט המשחק:

המשחק הינו פאזל בגודל $N \times N$ כאשר $N \in \{3, 4, 5\}$. את הלוח ממלאים $N^2 - 1$ קוביות אשר כל אחת מהן מייצגת חלק של תמונה. בתחילת המשחק, בלוח מוצגת לנו תמונה כלשהי מבולגנת והמטרה של השחקן היא לסדר את הלוח כך שהתמונה שתתקבל תהיה מסודרת והתא הריק יהיה בצד ימין התחתון של התמונה. סידור הלוח מתבצע ע"י הזזת הקוביות ימינה, שמאלה, למטה או למעלה. הזזת קובייה תתאפשר רק בתנאי שהמיקום אליו אנו מעוניינים להזיז אינו מכיל חלק של תמונה.

לדוגמא:



תרשים 1: לוח מבולגן בגודל 3X3

לאחר הזזת כמה מהקוביות נקבל:



תרשים 2: לוח בגודל 3X3 כמעט מסודר

לאחר הזזת הקובייה הימנית התחתונה שמאלה, נקבל את הפאזל הפתור:



תרשים 3: לוח מסודר בגודל 3X3

שימו לב שכאשר הצלחנו לפתור את הפאזל, אנחנו רואים את התמונה השלמה, ללא חלקים חסרים, כמו שמופיע בתרשים 4:



תרשים 4: לוח מסודר בגודל 3X3 ללא חלקים חסרים

התמונה להמחשה בלבד. אין צורך שהלוח יראה כך. ככל שהממשק יהיה יותר אטרקטיבי, כך תקבלו יותר נקודות.

2.0 תפריט ראשי:

המשחק מתחיל בתפריט ראשי שמכיל שני פריטים:

1. התחל משחק.
2. יציאה מהתוכנית.

2.1 מהלך המשחק:

כאשר המשתמש בוחר באופציה של התחלת משחק, יופיע לו חלון אשר ישאל אותו מה גודל הלוח ועם איזו תמונה הוא מעוניין לשחק. גדלי הלוח האפשריים יהיו 3X3, 4X4 או 5X5. (הסבר מפורט ב 2.2) תמונות אפשר לבחור באחת משלוש האפשרויות הבאות: (מפורט ב 2.2.2):

1. בחירה מאוסף תמונות קיים

2. הגרלת תמונה מאוסף תמונות קיים

3. טעינת תמונה חדשה (בנוסף)

לאחר שהמשתמש בחר גודל לוח ותמונה, הוא מועבר למסך המשחק.

- מסך המשחק יציג למשתמש את התמונה המעורבת, את כמות הצעדים שביצע ואת הזמן שעבר מתחילת המשחק.
- במהלך המשחק יוכל המשתמש להזיז את חלקי התמונה ימינה, שמאלה, למעלה ולמטה לפי החוקים המוגדרים ב 2.1.1.
- כאשר המשתמש ביצע את הצעד האחרון והשלים את התמונה, תוצג לו התמונה המלאה ללא החלק החסר, ביחד עם סיכום המשחק (כמות הזמן ששיחק + כמות הצעדים שביצע). שימו לב שבשלב זה לא תתאפשר למשתמש לבצע עוד צעדים.
- בכל שלב יש אפשרות לבטל את הפעולה האחרונה: ראו 2.1.2

2.1.1 חוקי הזזת החלקים

במהלך המשחק המשתמש יוכל להזיז את חלקי הפאזל למעלה, למטה, שמאלה וימינה ע"י:

1. לחיצה על חלק הפאזל אותו מעוניינים להזיז.

כאשר המשתמש בוחר באפשרות זו, במידה והחלק ניתן להזזה (כלומר נמצא ליד תא ריק בלוח), החלק יזוז לכיוון התא הריק. לדוגמא, בתרשים 2 המשתמש לחץ על החלק שנמצא בצד ימין התחתון בלוח והגיע למצב המתואר בתרשים 3.

2. לחיצה על מקשי החיצים במקלדת (←→↓↑).

כאשר המשתמש בוחר באפשרות זו, התוכנית תבדוק איזה מהחלקים ניתן להזיז לכיוון שנחלץ, ואם קיים חלק כזה, הוא יזוז לכיוון הרצוי. לדוגמא, בתרשים 2 המשתמש לחץ על החץ ← והגיע למצב המתואר בתרשים 3.

- בכל הזזת חלק על הלוח, כמות הצעדים המוצגת במסך תעלה ב-1 (שימו לב שלחיצה על חלק או על אחד ממקשי החיצים במקלדת שאינו גרם להזזה של חלק כלשהו בלוח - לא יגרם לשינוי בכמות הצעדים שהמשתמש ביצע)

2.1.2 פעולת UNDO

המשתמש יכול לבחור פעולת UNDO במהלך השלב. פעולה זו תבטל את הפעולה האחרונה שבוצעה. ניתן לבצע UNDO מספר לא מוגבל של פעמים (כל עוד יש פעולה שניתן לבטל). שימו לב שביצוע UNDO אינו מוריד או מעלה את כמות הצעדים שבוצעו.

דוגמת המחשה לפעולת UNDO:

- המשתמש ביצע פעולה 1.
- המשתמש ביצע פעולה 2.
- המשתמש לחץ UNDO – פעולה 2 בוטלה.
- המשתמש ביצע פעולה 3.
- המשתמש לחץ UNDO – פעולה 3 בוטלה.
- המשתמש לחץ UNDO – פעולה 1 בוטלה.

- את פעולת ה-UNDO ניתן יהיה לבצע בשני דרכים:
1. לחיצה על כפתור UNDO הנמצא על המסך.
 2. לחיצה על **Ctrl+Z** במקלדת.

2.2 טעינת וייצוג הלוחות

עם הפעלת התוכנית (הרצה של קובץ ה-jar), התוכנית תטען את הלוחות האפשריים מתוך קובץ CSV בשם boards.csv (אפשר לקרוא על קבצי CSV כאן). כאשר עבור כל גודל של לוח יכולים להופיע בקובץ מספר לוחות אפשריים.

רק לאחר הטעינה של הלוחות והתמונות לתוך מבנה נתונים מתאים, התוכנית תציג למשתמש את מסך התפריט הראשי (חשבו מה הסיבה לכך...).

כאשר המשתמש יבחר להתחיל משחק בגודל $N \times N$ כלשהו, התוכנית תגריל לוח מתאים מתוך אוסף הלוחות שנטען וזה יהיה הלוח של המשחק. במידה ולא קיים לוח בגודל המתאים, המערכת תציג על כך הודעה, ולא תאפשר להתחיל את המשחק.

פורמט קובץ הלוחות:

כל תא בלוח מייצג חלק של התמונה השלמה ומיוצג ע"י מספר 1 עד $N - 1$ כאשר לוח מסודר יכול את

הקוביות מסודרות לפי הסדר מימין לשמאל ומלמעלה למטה והקובייה התחתונה מצד ימין תהיה הקובייה הריקה.

לדוגמא:

1. לוח שאינו מסודר בגודל 3×3 יראה למשל כמו תרשים 1, ובמבנה הנתונים יהיה מיוצג כך:

1	0	2
7	6	3
5	4	8

2. לוח מסודר בגודל 3×3 יראה כמו בתרשים 3, ובמבנה הנתונים יהיה מיוצג כך:

1	2	3
4	5	6
7	8	0

כאשר 0 מייצג תא ריק בלוח.

קובץ הלוחות יהיה מיוצג ע"י קובץ CSV כך:

N_1
 $Board(N_1 \times N_1)$
 N_2
 $Board(N_2 \times N_2)$

...

כלומר כל ייצוג של לוח יתחיל בגודל של N ואחריו הלוח, כך שכל שורה בקובץ (לאחר N) תייצג שורה בלוח. לדוגמא:

3
5,0,1
2,7,8
4,6,3
4
9,5,1,8
2,7,4,3
13,0,10,12
14,11,6,15

בדוגמא הנ"ל הלוח הראשון בגודל 3×3 , השורה הראשונה מכילה 5,0,1 השורה השנייה מכילה 2,7,8 והשורה השלישית מכילה 4,6,3. לנוחיותכם, מצורף קובץ boards.csv כדי שתוכלו להתחיל לעבוד עליו. (בבדיקה שתבוצע אנו נטען קובץ קונפיגורציה אחר. על כך אין להכניס את הקונפיגורציה הנמצאת בקובץ hard-coded לתוך הקוד, אלא לטעון את הקובץ בצורה דינמית).

2.3 טעינת התמונות

כל תא בלוח יכול חלק מתוך תמונה אחת שלמה (חוץ מתא אחד שיהיה ריק), כאשר כל חלק מהתמונה יתוויג ע"י מספר המציין את מיקומו בלוח מסודר. כלומר החלק, שבתמונה מסודרת יהיה במיקום השמאלי העליון, יתוויג ע"י המספר 1. החלק, שבתמונה מסודרת יהיה במיקום האמצעי העליון, יתוויג ע"י המספר 2, וכך הלאה. התוכנית שלכם צריכה להכיל אוסף של תמונות איתם יהיה ניתן לשחק. אחרי בחירת הלוח מהשלב הקודם, המשתמש יתבקש לבחור את התמונה איתה הוא רוצה לשחק. כעת, כל חלק בתמונה יהיה ממוקם לפי הקונפיגורציה שנבחרה, כלומר החלק המתוויג ע"י המספר X יהיה ממוקם היכן שהמספר X נמצא בקונפיגורציה. את התמונה המשתמש יוכל לבחור בשלושה דרכים:

2.3.1. בחירה מאוסף התמונות הקיים.

יש להציג למשתמש בחלונות קטנים את התמונות הנמצאות באוסף התמונות. המשתמש יוכל לבחור בתמונה איתה הוא רוצה לשחק.

2.3.2. הגרלת תמונה מאוסף תמונות קיים.

המשתמש יוכל לבקש להגריל תמונה כלשהי מאוסף התמונות הקיים שאיתה הוא ישחק.

עבור 1 ו-2, מצורפים תמונות שתוכלו לעבוד איתם. מוזמנים להוסיף עוד תמונות כאוות נפשכם ☺

2.3.3. טעינת תמונה חדשה (בנוסף)

במידה והחלטתם לממש את הפונקציונלית הנ"ל, בבחירת התמונה המשתמש יוכל לטעון תמונה מהמחשב, והתוכנית תצטרך לחתוך את התמונה לחלקים, ולהציגה מעורבבת במשחק. הערות והנחות:

- אתם יכולים להניח שהאורך והרוחב של התמונה שאותה המשתמש יטען, שווים. כלומר לא ייתכן שהמשתמש יטען תמונה ארוכה שלא יהיה ניתן לחלק אותה בצורה שווה לקוביות.

2. בסיום המשחק, התמונה שהמשתמש טען אינה נשמרת. כלומר במידה והמשתמש ירצה להשתמש שוב באותה תמונה שכבר טען בעבר, הוא יצטרך לטעון אותה מחדש בתחילת המשחק.

2.4 הצגת מידע למשתמש :

במהלך המשחק יוצגו למשתמש, בנוסף ללוח עצמו, את הזמן שעבר מתחילת המשחק ואת כמות הצעדים שביצע.

בסיום משחק על המערכת להציג למשתמש בחלון חדש את המידע הבא:

- כמות הצעדים שביצע.
- הזמן שלקח לשחקן לסיים את המשחק.

בנוסף, למשתמש יופיעו האפשרויות הבאות:

- חזרה לתפריט הראשי.
- חזרה לבחירת לוח.

3 בונוסים (15 נק'):

3.1 טעינת תמונה חדשה:

ראו סעיף 2.2.2.3

3.2 יצירת לוח בצורה דינאמית

שימו לב: בונוס זה ניתן לממש רק לאחר מימוש הבונוס המוגדר בסעיף 3.1.

כאשר המשתמש בוחר באופציה של התחלת משחק, תוצג לו אפשרות לבחור לוח **בכל גודל** $N \times N$ כלשהו, כאשר $N \geq 2$. כאשר המשתמש יבחר בגודל כלשהו ויבחר את התמונה שאיתה הוא רוצה לשחק, התוכנית תייצר לוח בגודל המבוקש ותציג לו פאזל מבולגן.

שימו לב: לא כל פרמוטציה של קוביות על הלוח ניתנות לפיתרון, לדוגמא:

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	0

עם הלוח הנ"ל **לא ניתן** להגיע להגיע למצב בו הלוח פתור (קוביות 14 ו 15 התחלפו).

(אנקדוטה: ממציא המשחק, סאם לוי, הציג את הלוח הנ"ל והציע \$1000 למי שיצליח לפתור אותו. כמובן שאף אחד לא הגיע לקבל את הפרס...)

עליכם לכתוב אלגוריתם שייצר כל לוח בגודל $N \times N$ בצורה רנדומלית, כלומר אם המשתמש יבקש פעמיים לשחק עם לוח באותו הגודל, סביר שיקבל שני לוחות שונים, כלומר שמעורבבים בצורה שונה.

4 הגשה (חשוב):

החלון הראשי של התוכנית הגרפית צריך להרחיב את מחלקת JFrame (כפי שראינו בתרגולים). במקרה של אי בהירות בהגדרה של הלוגיקה של המשחק (מקרה קצה וכו'), אנא פרסמו שאלות בפורום באתר של הקורס ונשמח לענות עליהם בהקדם.

עליכם להגיש את הקבצים הבאים בתוך קובץ HW3_ID1_ID2.zip (כאשר ID1, ID2 הם תעודות זהות של המגישים מופרדים ב " ").

בנוסף, עליכם לעבוד לפי ה coding-standards המופיעים בדף ה Assignments.

בתוך קובץ ה-zip יופיעו **בדיוק** הדברים הבאים:

1. דיאגרמת מחלקות (UML) בתוך קובץ pdf בשם: "hw3uml.pdf".
2. קובץ jar בשם hw3.pdf.
3. הסבר על מימוש הבונוס (במידה ובחרתם לממש את בונוס 3.2 - מפורט בסעיף 5)

- **אנא בדקו** שאתם מצליחים להריץ את קובץ ה-jar שלכן מהקומנד ליין.
- **אנא ודאו שקוד המקור מופיע בתוך קובץ ה-jar (פתחו אותו עם תוכנת דחיסה)!**

הערה: כאשר אתם משתמשים בקבצי מדיה (תמונות ועוד), עליכם לוודא שקבצי המדיה נטענים כמו שצריך בהרצת ה-jar, אם אין באפשרותכם לטעון את הקבצים ישירות מתוך קובץ ה-jar ניתן לשים את הקבצים הנ"ל מחוץ ל-jar בתוך קובץ ה-zip.

5 מפתח ציון:

ניקוד	דרישות
16 נק'	דיאגרמת מחלקות (UML) בתוך קובץ pdf בשם: "hw3uml.pdf".
5 נק'	עיצוב המערכת לפי עקרונות תכנות מונחה עצמים
30 נק'	מהלך המשחק (כמתואר בדרישה 2.1): <ul style="list-style-type: none">• (15 נק') חוקי הזזת קוביות (כמתואר בדרישה 2.1.1).• (15 נק') פעולת UNDO (כמתואר בדרישה 2.1.2).

20 נק'	טעינת הלוחות והתמונות והצגתם (כמתואר בדרישה 2.2): <ul style="list-style-type: none"> • (15 נק') טעינה וייצוג הלוחות (כמתואר בדרישה 2.2.1). • (5 נק') טעינה וייצוג התמונות (כמתואר בדרישה 2.2.2).
14 נק'	הצגת מידע למשתמש (כמתואר בדרישה 2.3): <ul style="list-style-type: none"> • (7 נק') הצגת מידע למשתמש במהלך משחק (כמתואר בדרישה 2.3.1). • (7 נק') הצגת מידע למשתמש לאחר סיום משחק (כמתואר בדרישה 2.3.2).
15 נק'	אטרקטיביות ושימושיות כללית של המשחק (יצירתיות)
בונוס	
6 נק'	מימוש בונוס 3.1
9 נק'	מימוש בונוס 3.2 + כתיבת מסמך קצר המסביר כיצד מימשתם את האלגוריתם ליצירת הלוחות.

בהצלחה !