

עבודה 4 - עקרונות תכנות מונחה עצמים 2015

תאריך פרסום: 31/05/15 תאריך הגשה: 23:59 27/06/15

אחראי עבודה: Hadar Amran

1. הסבר כללי:

במשימה זו תצטרכו לממש גרסה פשוטה של המשחק Chicken Invaders 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 (ראה: סעיף 3). דרישות נוספות יכולות להעלות לכם את הציון כמפורט בסעיף הבנוס (ראה: סעיף 4). סך הציון בתרגיל יכול להגיע ל 115, כאשר 15 הם הנקודות של הבנוס. עליכם להתחיל את המטלה בקריאת מסמך זה, ניתוח של הבעיה ובבנייה של תרשים מחלקות ה (UML) המתאים, על פי הדרישות.

אם אתם לא מכירים את המשחק Chicken Invaders, ניתן לקרוא עליו בויקיפדיה.

כמו כן אתם יכולים לשחק בגרסה ה - 2,3 של המשחק בקישורים הבאים : (זהירות ממכר).

- <http://www.doublegames.name/swf/chicken-invaders2/chicken-invaders2.swf>
- <http://www.doublegames.com/swf/chicken-invaders-3-revenge-yolk-easter/chicken-invaders-3-revenge-yolk-easter.swf>

שימו לב!

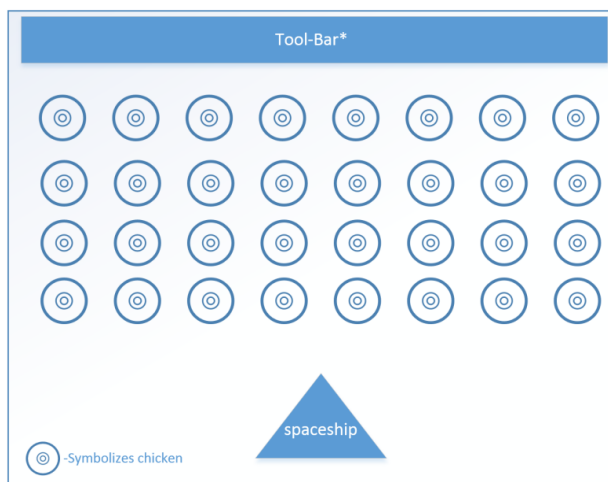
- הגרסה בעבודה זו שונה מהגרסה המקורית.
- אין צורך להכיר את המשחק בגרסה המקורית בכדי לבצע את המטלה.

משאבי תמונות/קול: לעבודה מצורף ארכיון Chicken_Invaders_resources.zip בו נמצאות תמונות של תרנגולות מסוגים שונים, תמונה של החללית, וקול, אשר יעזרו לכם בביצוע העבודה.

2. פירוט המשחק:

המשחק מורכב מ - 4 שלבים, כאשר כל שלב בנוי ממטריצה מסדר 4 שורות על 8 עמודות של תרנגולות (32 תרנגולות) מסוגים שונים ומחללית, ראה איור.

Tool-Bar* - מיקום סרגל הכלים נתון
לבחירתכם ולעיצובכם, אך עליו להכיל רכיבים גרפיים המציגים מידע למשתמש, (כמתואר בסעיף 2.4.1).



1 - יצוג גרפי של שלב במשחק

2.1 מהלך המשחק:

בתחילת המשחק מופיע למשתמש התפריט הראשי, בו יוכל לבחור להתחיל את המשחק בכל שלב i כלשהו שירצה , $1 \leq i \leq 4$, והניקוד ההתחלתי שלו הוא 0.

לאחר שהמשתמש בחר את בשלב i , הוא מתחיל לשחק בו:

- בכדי לעבור לשלב הבא על המשתמש להשמיד את כל התרנגולות במסך, בעזרת החללית. החללית תיצור יריות שונות להריגה של תרנגולות מסוגים שונים על פי חוקים מוגדרים (ראה סעיף 2.3).
- לאחר שהמשתמש השמיד את כל התרנגולות בשלב הנוכחי יחושב לו הניקוד עבור השלב הנוכחי על פי נוסחה (המתוארת בסעיף 2.4.2), ניקוד זה יתווסף לניקוד אותו צבר עד כה בשלבים הקודמים. התוצאות של השלב והתוצאה המצטברת יוצגו למשתמש, לפני המעבר לשלב הבא (או בסיום השלב האחרון).
- אם השלב אותו סיים הוא השלב הרביעי ($i = 4$) אזי המשתמש סיים את המשחק והוא יחזור לתפריט הראשי. אחרת ימשיך לשלב הבא ($i + 1$).

2.2 תיאור הרכיבים במשחק:

2.2.1 תרנגולות: קיימים סוגים שונים של תרנגולות

תרנגולות בצבעים –

# מס' סידורי	שם	תמונה
1	תרנגול אדום	
2	תרנגול כחול	
3	תרנגול צהוב	
4	תרנגול כתום	
5	תרנגול סגול	
6	תרנגול ירוק	





תרנגולות מיוחדים –

# מס' סידורי	שם	תמונה
7	תרנגול x	
8	תרנגול פלוס	
9	תרנגול עיגול	

תרנגולות בונוס (ראה: סעיף 4) -

# מס' סידורי	שם	תמונה
10	תרנגול שורה	
11	תרנגול עמודה	
12	תרנגול xor	
13	תרנגול שפעת העופות	

2.2.2 יריות: קיימים סוגים שונים של יריות:

שם/סוג	המחשה	מקש טעינה לחללית	KeyCode
ירייה שחורה		ספרה 1 : • במקלדת. • בפד.	e.getKeyCode()== • KeyEvent.VK_1 • KeyEvent.VK_NUMPAD1
ירייה אדומה		ספרה 2 : • במקלדת. • בפד.	e.getKeyCode()== • KeyEvent.VK_2 • KeyEvent.VK_NUMPAD2
ירייה כחולה		ספרה 3 : • במקלדת. • בפד.	e.getKeyCode()== • KeyEvent.VK_3 • KeyEvent.VK_NUMPAD3
ירייה צהובה		ספרה 4 : • במקלדת. • בפד.	e.getKeyCode()== • KeyEvent.VK_4 • KeyEvent.VK_NUMPAD4

ראו הסבר בסעיף הבא 2.2.3 בפעולה
"טעינה של סוג ירייה" של החללית



2.2.3 חללית:

לחללית יש מס' פעולות אפשריות:

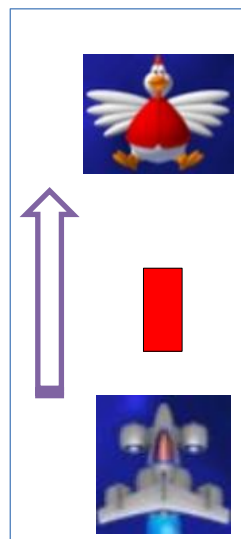
- יכולה לנוע ימינה (ציר ה-x חיובי), כאשר אסור לה לחרוג מהמסגרת של המשחק.
 - יכולה לנוע שמאלה (ציר ה-x שלילי), כאשר אסור לה לחרוג מהמסגרת של המשחק.
 - טעינה של סוג ירייה נוכחי:
- ישנם 4 סוגי יריות, והחללית יכולה לטעון את הירייה הנוכחית:

לחיצה על המקש 1 (בפד או במקלדת), מעדכנת את הירייה הנוכחית של החללית לשחורה.
לחיצה על המקש 2 (בפד או במקלדת), מעדכנת את הירייה הנוכחית של החללית לאדומה.
לחיצה על המקש 3 (בפד או במקלדת), מעדכנת את הירייה הנוכחית של החללית לכחולה.
לחיצה על המקש 4 (בפד או במקלדת), מעדכנת את הירייה הנוכחית של החללית לצהובה.
כאשר הירייה השחורה היא ברירת מחדל (default) בתחילתו של כל שלב.

- ירי של סוג הירייה הנוכחית:
בעת לחיצה על המקש רווח על החללית לבצע ירי מסוג של הירייה הנוכחית.

לדוגמא (ראה ציור מימין):














נניח שהירייה הנוכחית היא אדומה, אז בעת לחיצה על מקש הרווח נוצרת אנימציה של ירייה אדומה שמתקדמת בקו ישר כלפי מעלה לכיוון התרנגולות (ציר y שלילי).



2.3 תיאור חוקי ירייה בתרנגולות:

בעת מפגש של ירייה מסוג x שנוצרה על ידי החללית, לתרנגולת מסוג y נוצרים אירועים שונים. האירוע שנוצר בעת מפגש של תרנגולת וירייה תלוי ב $\langle x, y \rangle$ ומתואר בטבלה הבאה:

כל תא בטבלה מציג אילו תרנגולות **מתפוצצות** בעת מפגש ירייה ותרגול.

#	ירייה x \תרנגולת y	שחור	צהוב	כחול	אדום
1	אדום 	-----	2 תרנגולים כתומים בלוח כלשהם (אם קיימות).	2 תרנגולים סגולים כלשהם בלוח (אם קיימות)	התרנגולת y לבדה מתפוצצת.
2	כחול 	-----	2 תרנגולים ירוקים כלשהם (אם קיימות)	התרנגולת y לבדה מתפוצצת.	2 תרנגולים סגולים כלשהם בלוח (אם קיימות)
3	צהוב 	-----	התרנגולת y לבדה מתפוצצת.	2 תרנגולים ירוקים כלשהם (אם קיימות)	2 תרנגולים כתומים בלוח כלשהם (אם קיימות)
4	כתום 	-----	התרנגולת y לבדה מתפוצצת.	-----	התרנגולת y לבדה מתפוצצת.
5	סגול 	-----	-----	התרנגולת y לבדה מתפוצצת.	התרנגולת y לבדה מתפוצצת.
6	ירוק 	-----	התרנגולת y לבדה מתפוצצת.	התרנגולת y לבדה מתפוצצת.	-----
תרנגולות מיוחדים					
7	איקס 	התרנגולת y מתפוצצת וכל תרנגולת שנמצאות איתה באלכסון כלשהו מתפוצצת.	-----	-----	-----
8	פלוס 	התרנגולת y מתפוצצת וכל תרנגולת שנמצאות איתה בשורה או עמודה מתפוצצת	-----	-----	-----
9	עיגול 	התרנגולת y מתפוצצת וכל תרנגולת שנמצאות מסביבה במעגל (רדיוס תא אחד) מתפוצצת.	-----	-----	-----
תרנגולות בונוס (ראה: סעיף 4).					
10	עמודה 	התרנגולת y מתפוצצת וכל תרנגולת שנמצאות איתה בעמודה מתפוצצת.	-----	-----	-----
11	שורה 	התרנגולת y מתפוצצת וכל תרנגולת שנמצאות איתה בשורה מתפוצצת.	-----	-----	-----
12		התרנגולת y מתפוצצת וכל תרנגולת שנמצאת בעמודה או בשורה או מסביבה במעגל (רדיוס תא אחד) מתפוצצת.	-----	-----	-----
13		כל התרנגולות במסך מתפוצצות.	-----	-----	-----

2.4 הצגת מידע למשתמש בשלב ובסיומו:

2.4.1 Toolbar (מידע על מצבו של השחקן ברגע נתון בשלב):

בכל שלב יוצג המידע הבא למשתמש בתוך ה Toolbar ,

- זמן מתחילת השלב (Time): הזמן בשניות מתחילת השלב הנוכחי. כלומר על הסטופר בתחילת השלב **להתאפס**, ולרוץ במרווח של **שנייה** (כל שנייה להתעדכן לערך הבא).
- כמות יריות (Shots): מספר היריות שנורו מתחילת השלב הנוכחי על ידי השחקן. כלומר על כמות יריות להתאפס, וכל ירייה של השחקן מס' היריות עולה האחד.

2.4.2 ניקוד בסיום שלב/משחק:

בסיום כל שלב (כולל השלב האחרון) **ולפני** תחילתו של השלב הבא, על המערכת להציג למשתמש את המידע הבא:

- זמן סיום השלב הנוכחי (level time): הזמן בשניות לסיום השלב הנוכחי.
- כמות יריות בשלב הנוכחי (level Shots): מספר היריות הכולל שנורו בשלב הנוכחי על ידי השחקן.
- ניקוד של השלב הנוכחי (level Score): הניקוד של השלב הנוכחי, יחושב על ידי level Shots , level time על פי הנוסחה הבא:
$$\text{level_Score} = \max \{ 0, 600 - [\text{level_Shots}] \cdot 10 - [\text{level_time}] \}$$

לדוגמא :

level_Score	level_Shots	level_time (seconds)
380	10	120
0	50	200
250	5	300

- ניקוד מצטבר/סופי (total Score): סכום כל ה level Score בכל השלבים עד כה (כולל השלב הנוכחי שסיימו). לדוגמא:
נניח שהשחקן סיים כרגע את שלב 3 עם ניקוד 300, ובשלב הראשון והשני קיבל ניקוד 200 100 בהתאמה.
אזי $\text{total_Score} = 100 + 200 + 300 = 600$.

3 חובת מימוש/דרישות פונקצינאליות:

כלל אין אנו מתעריבים בעיצוב המחלקות ובאפשרות לבחור איזה עיצוב שתמצאו למימוש. הדרישות היחידות שלנו הן כדלקמן:

3.1 שלב התוכנית ושימוש בתבנית המארח (Visitor pattern):

על התוכנית שלכם להכיל את המחלקות הבאות והיחסים ביניהם.

אין באמור בעבודה זו כדי לפגוע בתרגולות.

3.1.1 : Visitor

ממשק המכיל חתימות לפונקציות בהתאם לתבנית המארח (Visitor pattern).

3.1.2 : Chicken

מחלקה (רגילה, ממשק או מחלקה אבסטראקטית לבחירתכם), אשר מממשת (או מרחיבה) את הממשק "Visitor".

3.1.2 : Shot

ממשק או **מחלקה אבסטראקטית** לבחירתכם, כך שמכיל את הפונקציה:

• עבור **ממשק**:

Public void shooting(Visitor v);

• עבור **מחלקה אבסטראקטית**:

Public abstract void shooting(Visitor v); /*Symbolizes the method "accept" in Visitor pattern*/

לוגיקת מפגש ירייה ותרנגולת (תבנית המארח):

על הפונקצייה shooting לבצע את מימוש הלוגיקה של מפגש בין ירייה לתרנגולת (על פי החוקים המוצגים ב 2.3). במימוש זה **אסור** להשתמש ב casting או באופרטור instanceof. כלומר עליכם להשתמש בתבנית המארח (Visitor pattern).

רמז:

כל מחלקה היורשת מ Shot **חייבת** לדרוס את shooting באופן הבא,

```
Public void shooting(Visitor v){ /*Symbolizes the method "accept" in Visitor pattern*/
    v.visit(this);
}
```

חובת דריסה זו, מאפיינת את חתימות הפונקציות בממשק **Visitor**.

3.2 מימוש של שלבים והצגתם:

3.2.1 עליכם לממש לפחות את 4 השלבים של המשחק על פי המטריצות הבאות. כאשר כל שלב מוגדר על ידי מטריצה 4×8 ובכל תא מכיל מס' סידורי של תרנגולת (לפי הגדרתן בחלק 2.2.1).

שלב ראשון (1)	שלב שני (2)
$\begin{pmatrix} 1 & 2 & 3 & 1 & 1 & 3 & 2 & 1 \\ 3 & 1 & 2 & 2 & 2 & 2 & 1 & 3 \\ 2 & 3 & 1 & 3 & 3 & 1 & 3 & 2 \\ 1 & 2 & 3 & 4 & 4 & 3 & 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 5 & 5 & 5 & 6 & 6 & 4 & 4 & 4 \\ 3 & 9 & 1 & 1 & 1 & 1 & 9 & 3 \\ 3 & 1 & 2 & 2 & 2 & 2 & 1 & 3 \\ 6 & 5 & 4 & 3 & 3 & 4 & 5 & 6 \end{pmatrix}$
שלב שלישי (3)	שלב רביעי (4)
$\begin{pmatrix} 5 & 5 & 5 & 3 & 3 & 4 & 4 & 4 \\ 5 & 8 & 5 & 2 & 2 & 6 & 6 & 6 \\ 5 & 5 & 5 & 1 & 1 & 6 & 8 & 6 \\ 4 & 4 & 4 & 8 & 4 & 6 & 6 & 6 \end{pmatrix}$	$\begin{pmatrix} 5 & 5 & 4 & 4 & 5 & 5 & 4 & 4 \\ 6 & 7 & 6 & 7 & 6 & 7 & 6 & 7 \\ 4 & 3 & 5 & 2 & 4 & 3 & 5 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

3.2.2 הצגת בחירה של השלבים בתפריט הראשי:

על מנת שניתן יהיה לבדוק את כל 4 השלבים בנפרד, עליכם לממש בתפריט הראשי אפשרות לשחקן להתחיל מאיזה שלב i התחלתי שירצה ($i \in \{1, 2, 3, 4\}$).
הניקוד הכללי (total Score) בתחילת השלב ה- i הינו 0 (ראה 2.4.2).

3.3 הצגת מידע למשתמש:

עליכם להציג מידע למשתמש בסיום ובמהלך כל שלב כמתואר בחלק 2.4.

3.4 מימוש התנהגות החללית:

עליכם לממש את ההתנהגות של החללית (לנוע ימינה/שמאלה, החלפת סוג ירייה על ידי מקשי המקלדת) כמתואר בחלק 2.2.3.

4 בונוס (15 נק'):

4.1 הוספה של תרגולות נוספים (6 נק'):

באפשרותכם להוסיף עוד 4 תרגולות בעלות המס' הסידורי 10,11,12,13 כפי שהוגדרו ב 2.2.1. כאשר חלוקת הניקוד מתבצעת כך:

- הוספת תרגולות 10,11 מהווה 1 נק' כל אחת.
- הוספת תרגולות 12,13 מהווה 2 נק' כל אחת.

בכדי שתוכלו לממש את התרגולות האלו, תצטרכו להוסיף שלבים נוספים לבחירתכם ולהוסיף את התרגולות האלו אל אותם שלבים.

4.2 הוספה של טבלת ניצחונות אחרונים (9 נק'):

בתפריט הראשי, יש אפשרות לשחקן לצפות (לחיצה על כפתור), בטבלת של 5 הניצחונות האחרונים של שחקנים (ניצחון אחרון = סיום שלב אחרון).
כלומר הטבלה מכילה לכל היותר 5 רשומות של 5 השחקנים האחרונים אשר סיימו את המשחק (כלומר סיימו את השלב האחרון) עם המידע הבא:

- שם שחקן : שם של השחקן שסיים את השלב האחרון, למעשה בסיום של השלב האחרון נפתח חלון לשחקן להכניס את שמו, על מנת שנוכל להכניסו לטבלה.
- ניקוד מצטבר: total Score בסיום השלב האחרון כפי שמחושב בחלק 2.4.2.
- זמן סיום: מתי סיים השחקן את השלב האחרון.

אבחנה: יתכן האפשרות שהשחקן בחר בתפריט הראשי לשחק בשלב האחרון (3.2.2), ועדיין אנחנו נכניס אותו לתוך הטבלה.

שימו לב: הטבלה צריכה להיות מקודדת לתוך קובץ בפורמט שתבחרו. כך שבטעינת המשחק, נוכל לראות את השחקנים האחרונים ששיחקו ובפרט את אלו אשר שיחקו בטעינות קודמות של המשחק.

4.3 דוח בונוס:

שימו לב! חובה לכל מי שממש את החלק של הבונוס להוסיף את הקובץ bonus.txt ולסמן בתוכו את המידע על מימוש הבונוס כפי שמתואר בקובץ זה המצורף לעבודה.

5 המלצות:

- ירייה סדרתית: בעת ירייה של החללית (לחיצה על **מקש רווח**) ירייה חדשה תִּצָּר **אם"ם** לא קיימת כבר ירייה הנמצאת על המסך.
כלומר בכל רגע נתון במשחק רק **ירייה אחת** נמצאת על המסך בתנועה.
תנאי זה **מונע** את הטיפול של מס' יריות על המסך במקביל (multithreading).
- משאבים: לעבודה מצורף ארכיון Chicken_Invaders_resources.zip בו נמצאות תמונות של תרנגולות מסוגים שונים, תמונה של החללית, וקול, אשר יעזרו לכם בביצוע העבודה.
- דוגמא: מצורפת תוכנית גרפית GUI_Task4_Example.jar , המכילה גם את קוד המקור לתוכנית (עליכם לחלץ את התוכן של ה jar וכך תוכלו לראות את הקוד).
תוכנית זו נכתבה במיוחד כדוגמא למטלה זו ואין חובה להסתכל בה.
כמו כן תוכנית זו מכילה פונקציות/דוגמאות שימושיות לעבודה 4.
אין אנו לוקחים אחריות על באגים בתוכנית הגרפית הנ"ל,
אז ישתמשו בה בחוכמה.

6 הגשה:

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

עליכם להגיש את הקבצים הבאים בתוך קובץ X_Y.zip (כאשר X,Y הם תעודות זהות של המגישים מופרדים ב " _ ") ללא תיקיות:

1. [15 נקודות]. דיאגרמת מחלקות (UML) בתוך קובץ pdf בשם: "hw4uml.pdf".
2. [85 נקודות]. **קוד המקור** והקוד המקומפל בתוך קובץ jar בשם hw4.pdf.
3. [15 נקודות **בבונוס**] קובץ bonus.txt עם כל המידע על מימוש הבונוס (כמתואר 4.3).

הערה : לגבי קבצי 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 .

7 מפתח ציון:

ניקוד	דרישות
15 נק'	דיאגרמת מחלקות (UML) בתוך קובץ pdf בשם: "hw4uml.pdf".
15 נק'	מימוש של התנהגות החללית (כמתואר בדרישה 3.4)
20 נק'	שימוש בתבנית המארח ובשלב התוכנית (כמתואר בדרישה 3.1)
15 נק'	מימוש של חוקי ירייה לתרנגולת (כמתואר בדרישה 3.5)
15 נק'	הצגה של מידע למשתמש בסיום של שלב ובמהלכו (כמתואר בדרישה 3.3)
20 נק'	מימוש של כל 4 השלבים (4 נק' לשלב) והצגתם לשחקן בתפריט הראשי (4 נק')
	בבונוס
6 נק'	הוספה של תרנגולות נוספות (כמתואר ב 4.1).
9 נק'	הוספה של טבלאת ניצחונות אחרונים (כמתואר ב 4.2).

בהצלחה !