

# תוכנה 1 – אוניברסיטת תל אביב – סמסטר ב' 2024

## תרגיל מספר 4

להגשה עד 19.12.24 בשעה 23:59

### הנחיות כלליות:

- קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.  
את התרגיל הבא צריך להגיש באופן הבא:
  - הגשה במערכת ה-Git. יש להתקין Git, להרשם ל-GitHub וליצור SSH-key על המחשב האישי שלכם, על פי ההנחיות בתירגול 1.  
לאחר מכן, צרו את ה repository שלכם מתוך הקישור הבא:  
<https://classroom.github.com/a/I0Y8mGJc>  
יש לוודא שבתיקיית הגיט שלכם נמצאים הקבצים הבאים:  
a. קובץ פרטים אישיים בשם details.txt המכיל את שם המשתמש שלכם ב Moodle ואת מספר תעודת הזהות שלכם.
- שימו לב: אם בדף הפרויקט בגיט הוא לא מופיע, העתיקו אותו מתרגיל בית קודם.**
- b. קבצי ה- java של התוכניות אותם התבקשתם לממש.
  - הגשה במערכת ה Moodle (<http://moodle.tau.ac.il/>): עליכם להגיש את קובץ הטקסט assignment.txt ובו קישור לgit repository האישי שלכם. הקובץ צריך להכיל שורה אחת בדיוק, ללא מלל נוסף. לדוגמה, עבור תרגיל 1 הקובץ יכיל את השורה הבאה, כשבמקום githubUser יופיע המשתמש שלכם ב github:  
<https://github.com/software1course2025a/hw-1-githubUser.git>
- בדקו את עצמכם:  
הורידו את התרגיל מחדש על המחשב שלכם, רצוי במיקום שונה מהמיקום עליו עבדתם.

בתרגיל זה תעבדו על תוספת קוד שמערב מנשקים לחבילת תוכנה קיימת (משחק פקמן).

0. ודאו שביצעתם את הוראות ההתקנה של **LWJGL** שמופיעות במדריך במודל.

ניתן גם לצפות בסרטון ההדרכה: <https://youtu.be/LrZSW2qR1Tw>

בחבילה `il.ac.tau.cs.software1.inventory` (מעטה ואילך בהתייחסות לחבילות, נשמיט את הקידומת) מופיעים שני מנשקים: `Inventory`, `Collectible`. כאשר `Inventory` מייצג מנשק עבור אוסף של `Collectible`-ים.

בתרגיל זה, נממש את שני הקונספטים האלו, כך שפקמן, במקום לאכול ולהרוס את הבונוסים - יאפסן אותם בצד (כדי שיוכל למכור אותם אחר כך). שימו לב להוסיף בתחילת קבצי `java` משפטי `import` רלוונטיים, בהתאם למחלקות בהן אתם משתמשים (סביבת אקליפס תציע לעזור לכם עם זה).

1. תחת החבילה `inventory`, נוסיף מחלקה חדשה בשם **`NaiveInventory`** שתממש את המנשק `Inventory`. היא תכיל מערך של `Collectibles` (שאוּלִי יהיה גדול יותר מכמות החפצים בפועל), ומשתנה מטיפוס `int` שסופר כמה חפצים יש בפועל. הגדירו גם קבוע, בראש המחלקה:

```
private final int INITIAL_CAPACITY = 100
```

באתחול `Inventory`, גודל המערך יהיה `INITIAL_CAPACITY`. הפונקציה `collect` של המנשק תנסה להוסיף את החפץ החדש למערך. אם כמות האיברים שווה לגודל המערך, ניצור מערך חדש שגודלו יהיה כפול מגודל המערך הקודם (נעתיק את האיברים שהיו קודם). ממשו גם את שאר המתודות של המנשק `Inventory`, בהתאם לתיאור שלהן בקובץ המנשק. בנוסף ממשו מתודה בשם `getCollArr` שמחזירה את המערך של `Collectibles` השמור בשדה במחלקה.

2. תחת החבילה `inventory`, צרו מחלקה חדשה בשם **`InventoryComponent`** שתממש את המנשק `IComponent` שנמצא בחבילה `components`. במחלקה זו יהיה שדה מטיפוס `Inventory`, שיאותחל להיות אובייקט מהמחלקה `NaiveInventory`. כמו כן, הוסיפו את הקבוע הבא:

```
private static final Vector2 INVENTORY_POSITION = new Vector2(-100.0f, -100.0f);
```

תידרשו לעשות `import` ל-`Vector2` שיושב במחלקה `math` בפרוייקט זה.

ממשו שני `getter`ים: `getInventory` שיחזיר את השדה מטיפוס `Inventory`, ו-`getInventoryPosition` אשר יחזיר את הקבוע.

3. תחת החבילה `pacman`, במחלקה **`PacmanPlayer`**, הוסיפו שדה פרטי מטיפוס `InventoryComponent`, ואתחלו אותו בהתאם.

4. הסתכלו על המחלקה **`PacmanBonus`** שמופיעה בחבילה `pacman`. מחלקה זו מממשת את המנשק `Collectible`. מחלקה זו הינה מחלקה אבסטרקטית (בהגדרת המחלקה מופיעה המילה `abstract`) -

נושא שילמד בהרצאות הקרובות - ולכן לא ניתן לייצר אובייקטים מטיפוס זה. מחלקה אבסטרקטית לא חייבת לממש את מתודות הממשק כי ממילא לא ניתן לייצר אובייקטים מהטיפוס של המחלקה. את המתודות של הממשק נממש במחלקה במחלקה **PacmanBonusDot**: המשקל של כל בונוס נקודה הוא 1.0 והמחיר של כל בונוס נקודה הוא 12.0.

**הערה:** שימו לב שבין המחלקה **PacmanBonusDot** למחלקה **PacmanBonus** מתקיים קשר של ירושה - נכיר קשר זה בשיעורים הקרובים. כרגע מבחינתכם חשוב לדעת ש **PacmanBonusDot** היא "סוג של" **PacmanBonus**.

5. תחת החבילה **pacman**, בפונקציה **onHit** של המחלקה **PacmanPlayer**: נשים לב שאנחנו בודקים במה השחקן פגע. אם השחקן פגע בבונוס - אנחנו נצבור את הניקוד שהבונוס העניק לנו, ואז נהרוס את הבונוס (נקרא למתודה **destroy** של בונוס). נבצע את השינוי הבא. במקום פעולות אלה נבצע את הפעולות הבאות:

a. נשנה את המיקום של הבונוס להיות ב **INVENTORY\_POSITION** (רמז: תרצו להשתמש ב: **position** ב: **bonus.getTransformComponent().position**)  
**העשרה:** הקצאות זיכרון (כולל יצירה או מחיקה תכופה של אובייקטים) היא פעולה יחסית יקרה, ולכן במשחקים רבים נוהגים להשתמש באותו הטריק של להחביא את האובייקטים במקום שנסתר מהשחקן. למשל:  
<https://youtu.be/yK1mRhyeCzA?si=Fqp3llsWLv2Oxs2X&t=214>  
<https://www.pcgamer.com/heres-whats-happening-inside-fallout-3s-metro-train>

b. נאסוף את הבונוס (**collect**) אל תוך ה**Inventory** שב**InventoryComponent** של השחקן.  
c. נצבור את הניקוד שהבונוס העניק לנו (ולא נהרוס את הבונוס!).

6. פקמן שלנו לא יכול לסחוב יותר מ-10.0 קילוגרם! נממש את התכונה הזו שלו.  
הוסיפו למחלקה **InventoryComponent** שדה של **maxWeight** (עם **getters** ו**setters** מתאימים). הגדירו בראש המחלקה קבוע מתאים (**INITIAL\_MAX\_WEIGHT = 10**), ואתחלו את **maxWeight** בהתאם. כעת, בצעו את הקוד שבסעיף 5 (ההתנהגות ב **onHit**) אם ורק אם לאחר האיסוף, לא נחרג מ**maxWeight**.  
שימו לב, מותר ש **maxWeight == totalWeight** אחרי האיסוף, אבל אסור לחרוג מ**maxWeight**. כמובן שאם לא אספנו את הבונוס - גם לא נצבור עבורו ניקוד ולא נהרוס אותו.

7. הריצו את המשחק ע"י הרצת פונקציית **main** שנמצאת במחלקה **PacmanGameApplication**. בדקו האם פקמן אכן אוסף רק עשרה בונוסים, ושה**score** לא חורג מ-100.  
שנו את המשקל של הבונוס כרצונכם ובדקו אם המשחק מתנהג כמו שציפיתם גם לאחר השינוי.

## בהצלחה !