תוכנה 1 – אוניברסיטת תל אביב – סמסטר ב' 2024

תרגיל מספר 7

להגשה עד 28.01.25 בשעה 23:59

הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.
את התרגיל הבא צריך להגיש באופן הבא:

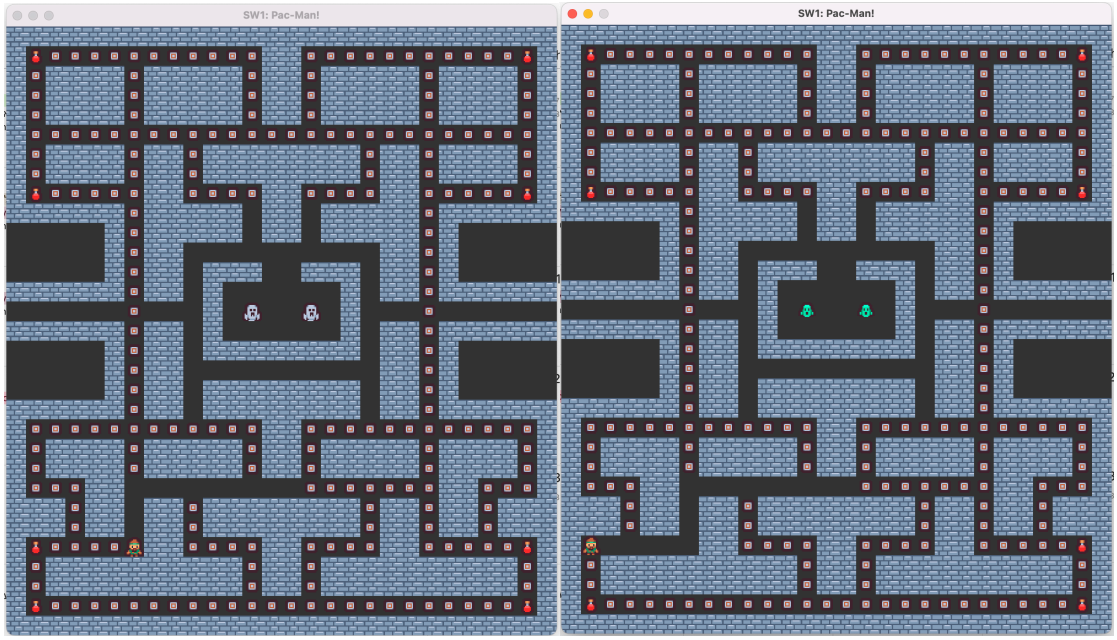
- הגשה במערכת ה-Git. יש להתקין Git, להרשם ל-GitHub וליצור SSH-key על המחשב האישי שלכם, על פי ההנחיות בתירגול 1.
  לאחר מכן, צרו את ה repository שלכם מתוך הקישור הבא:
  https://classroom.github.com/a/gaKA0DKn
  יש לוודא שבתיקיית הגיט שלכם נמצאים הקבצים הבאים:
  a. קובץ פרטים אישיים בשם details.txt המכיל 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 לrepository git 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 נוסיף מנהל אירועים (קרוי גם Observer design pattern) למשחק הפקמן.
נוספו למשחק שני אובייקטים חדשים: בונוס שיקוי, ורוח. המטרה: כאשר השחקן אוסף את השיקוי, כל רוח אוטומטית תהפוך ל"מפוחדת".
דרך אחת ומסורבלת לבצע את זה, היא בפונקציית הupdate של הרוח: נעבור על כל האובייקטים מטיפוס "בונוס שיקוי" (כיצד נשיג אותם?), ונבדוק האם אחד מהם נאסף לאחרונה. אם כן – נעדכן את הרוח. חשבו מדוע הדרך הזו מסורבלת מדי ולאילו בעיות מימוש שכזה עשוי לגרום בהמשך.
דרך נוספת ויותר מקובלת לחבר בין השניים היא דרך מחלקה צד ג', שהיא "מנהל אירועים".
המחלקה PacmanGhost לא צריכה להכיר שקיים טיפוס PachmanBonusPotion ולהפך. מחלקת מנהל האירועים בוודאי לא מתייחסת במפורש לאף טיפוס ספיציפי.
החיבור קורה באופן הבא: מחלקת הבונוס מפרסמת (notify) הודעת "potionCollected" כאשר השחקן אוסף את הבונוס.
בבנאי של הרוח, כל רוח תירשם (subscribe) לאירוע "potionCollected" עם מצביע לפונקציה (callback), כלומר מנשק פונקציונאלי – כפי שלמדנו בשיעור.
כאשר אירוע קורה, מנהל האירועים עובר על כל האובייקטים שנרשמו לאותו אירוע, וקורא לcallback של אותם האובייקטים. כך אנחנו קושרים בין שני אירועים, בלי לייצר התייחסות מפורשת לטיפוס המחלקה אחת של השניה.
שימו לב שזו רק דרך אחת לממש את הObserver design pattern, ישנן דרכים נוספות.



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.core (מעתה ואילך בהתייחסות לחבילות, נשמיט את הקידומת) נוספו המחלקות EventData, EventManager, EventSubscription והמנשק IEventCallback.

בחבילה pacman, הוספנו את המחלקות PacmanBonusPotion וPacmanGhost, ועדכנו את הObject factory וקורא הסצנות כדי שייצר בונוס שיקוי ורוח, בהתאמה.

בתרגיל זה, נשלים את המימוש של המחלקה EventManager, ובעצם נממש את הקונספט של ה observer pattern.
לאחר מכן, בעזרת EventManager נחבר בין אירוע האיסוף של בונוס השיקוי, לפעולה המתאימה של הרוח. לפני תחילת התרגיל, עברו על כל המחלקות והמנשקים שציינו.

1. המחלקה **EventManager** צריכה להיות *Singleton*, בכדי שנוכל לגשת למופע (היחיד) שלה מכל מחלקה אחרת במשחק. החליפו את הקוד /* Q1 */ במימוש ה Singleton שראינו *בתרגול 4*.
גישה למופע של EventManager יעשה באמצעות הפונקציה (הפומבית) ()getInstance.

2. ממשו את מתודת subscribe של **EventManager**.
המתודה מקבל אירוע (מחרוזת), אובייקט ש"נרשם" לאירוע ומצביע לפונקציה (מנשק פונקציונלי), ומעדכן EventSubscription חדש במפת ה subscribers.
רמז: מה קורה כאשר עושים subscribe לאירוע חדש בפעם הראשונה?

3. ממשו את מתודת notifyEvent של **EventManager**.
המתודה מקבלת מחרוזת אירוע, אובייקט שאחראי על הדיווח עצמו, ומשתנה אופציונאלי data (להעברה של מידע נוסף יחד עם האירוע). לכל אובייקט שנרשם לאירוע הנתון, נקרא למתודת call של ה callback המתאים לו. שימו לב שמתודת call מקבלת ארגומנט יחיד, מטיפוס EventData.
שימו לב: ייתכן ואף סביר כי נדווח על אירוע גם אם אף אובייקט אחר לא נרשם לעדכונים שלו.

4. ממשו את מתודת onHit של **PacmanBonusPotion**.
המתודה תדווח אירוע *"potionCollected"* אם האובייקט other שהבונוס התנגש איתו הוא השחקן (מטיפוס PacmanPlayer). ה publisher של האירוע הזה הוא כמובן this, כלומר הבונוס עצמו.
אין צורך להעביר מידע נוסף יחד עם האירוע, כלומר data יהיה null.

5. כעת, בבנאי של **PacmanGhost**, נירשם לאירוע *"potionCollected"*.
תחילה, נדפיס על המסך את המחרוזת הבאה:
"Collected potion @ XXX"
כאשר במקום XXX יופיע ה position של ה publisher של האירוע. כדאי להשתמש במתודת toString של המחלקה Vector2.
לאחר מכן, אם הרוח עוד לא מפוחדת – נהפוך אותה למפוחדת, ולהפך.[1]
את ה callback כדאי (אבל לא חייב) לממש בעזרת ביטוי למבדה.
פלט תקין לדוגמא: `Collected potion @ (-0.9285714, 0.4193548)`

6. הריצו את המשחק ע"י הרצת פונקציית main שנמצאת במחלקה **PacmanGameApplication**. בדקו האם איסוף בונוס השיקוי אכן משנה את הרוח לסירוגין – מלא מפוחדת למפוחדת, ולהיפך.

בהצלחה !

[1] זאת כמובן בשונה מהמשחק המקורי, בו איסוף של הבונוס הופך את הרוח למפוחדת, והיא חוזרת בעצמה למצב לא מפוחד תוך כמה זמן. איסוף של בונוס נוסף בזמן שהיא מפוחדת מאריך את הזמן. בתור אתגר (לא קשה), חישבו כיצד הייתם מממשים את ההתנהגות הזו בקוד של הפקמן שלנו.