

תכנות מונחה עצמים תרגיל 3.1

בתרגיל זה עליכם לממש משחק דמוי "[Arkanoid](#)". לצורך משימה זו, אנחנו מספקים לכם מנוע - משחק בדמות הספריה DanoGameLab. [כפי שניתן לראות בסרטוני הקורס](#), בעזרת ספריה זו ניתן ליצור עצמים (אובייקטים) של המשחק שאותם ינהל המנוע בעזרת ה-GameManager. בתרגיל זה אתם תתכנתו את המשחק באמצעות המנוע הזה.

הוראות המשחק

את המטקה, הכדור, הלבנים, הרקע והקירות יש לבנות לפי ההוראות בסרטונים באתר [הקמפוס](#). במשחק יהיו 56 לבנים, 8 שורות של לבנים ו-7 לבנים בכל שורה.

- תנאי ההפסד במשחק יהיה פסילות מרובות. כלומר לאחר 3 פסילות, המשחק יודיע על הפסד.
- תנאי הניצחון במשחק יהיה לנקות את כל הלבנים מהמסך.
- לחיצה על כפתור 'W' תסיים את המשחק עם תנאי הניצחון (אין צורך לנקות את הלבנים מהמסך).
- יש להציג בכל רגע נתון בצורה טקסטואלית ובעזרת אייקון heart.png את כמות הפסילות שנותרו.
- עבור התצוגה הטקסטואלית, יש להשתמש במחלקה java.awt.Color.
 - אם מספר הפסילות הנותרות הוא 3, המונה יהיה בצבע ירוק (Color.green)
 - אם מספר הפסילות הנותרות הוא 2, המונה יהיה בצבע צהוב (Color.yellow)
 - אם מספר הפסילות הנותרות הוא 1, המונה יהיה בצבע אדום (Color.red)
- ניתן לשנות את צבע הטקסט ע"י שימוש במתודה setColor() של המחלקה TextRenderable.
- מקשי המשחק השולטים בפדאל הינם החצים שמאלה וימינה.

חומרים מסופקים

מסופק לכם לצורך מימוש המשחק קבצי assets. תיקיית ה-assets מכילה אוסף של קבצי תמונה לצורך רינדור אובייקטי המשחק – כדור, לבנים, מטקה, וכו'.

מבנה הספרייה של התרגיל יראה בדיוק כך, ויכלול את כל הקבצים המתוארים כאן ורק אותם:

```
./src
./src/BrickerGameManager.java
./src/brick_strategies
./src/brick_strategies/CollisionStrategy.java
./src/gameobjects
./src/gameobjects/Ball.java
./src/gameobjects/Brick.java
./src/gameobjects/GraphicLifeCounter.java
./src/gameobjects/NumericLifeCounter.java
./src/gameobjects/Paddle.java

./assets/
./assets/assets_files.txt
./assets/Attribution.txt
./assets/ball.png
./assets/blop_cut_silenced.wav
./assets/botBad.png
./assets/botGood.png
./assets/brick.png
./assets/Bubble5_4.wav
./assets/buffNarrow.png
./assets/buffWiden.png
./assets/DARK_BG2_small.jpeg
./assets/gravity.png
./assets/heart.png
./assets/mockBall.png
./assets/paddle.png
./assets/quicken.png
./assets/slow.png
```

מחלקות למימוש

ה-API המפורט כאן קיים בקבצי javadoc הנמצאים במודל, יש לפתור את התרגיל בהתאם להגבלות ה-API.

1. המחלקה Ball

- מחלקה זו יורשת מ-GameObject (המוגדרת בספריית DanoGameLab) ומתארת את הכדור במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה OnCollisionEnter, אשר משנה את כיוון הכדור לפי הכיוון ההפוך לנורמה של ה-collision המתקבל (ע"י collision.getNormal), ולאחר מכן משמיעה את הצליל המתקבל בבנאי.

2. המחלקה Paddle

- מחלקה זו יורשת מ-GameObject, ומתארת את המטקה במשחק.
במחלקה זו עליכם לממש:
- 1) הבנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה update (הנקראת בכל frame במשחק), ומעדכנת את מיקום המטקה בהתאם למקשי השחקן.

3. המחלקה Brick

- מחלקה זו יורשת מ-GameObject, ומתארת את הלבנים במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה OnCollisionEnter, אשר עליה להפעיל את collisionStrategy (יתואר בהמשך).

4. המחלקה GraphicLifeCounter

- מחלקה זו יורשת מ-GameObject, ומתארת את הלב (כמות הפסילות הגרפית) במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה update, אשר מעדכנת את כמות האייקונים של הלב המופיעים על המסך בהתאם לכמות הפסילות שונתרו.

5. המחלקה NumericLifecounter

- מחלקה זו יורשת מ-GameObject, ומתארת את מונה הפסילות הטקסטואלי במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה update, אשר מעדכנת את המונה הטקסטואלי (במספר ובצבע כפי שמתואר לעיל).

6. המחלקה CollisionStrategy

- מחלקה זו מתארת את אסטרטגיית התנגשות בלבנה במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) פונקציה OnCollision, אשר עליה למחוק את הלבנה בה התנגש הכדור מהמשחק, ולעדכן את מונה הלבנים.

7. המחלקה BrickerGameManager

- מחלקה זו יורשת מ-GameManager, והינה המחלקה הראשית במשחק.
במחלקה זו עליכם לממש:
- 1) בנאי המקבל את הפרמטרים המתוארים ב-API.
 - 2) הפונקציה initializeGame אשר מאתחלת את כל האובייקטים הנחוצים למשחק.
 - 3) את הפונקציה update אשר בודקת את מצב הכדור (האם קרתה פסילה ואיפוס מיקום הכדור),

ו-win condition:

- אם כמות הפסילות ירדה ל-0 (כלומר קרו 3 פסילות), המשחק נגמר ותוצג הודעת הפסד.
- אם כמות הלבנים ירדה ל-0, המשחק נגמר ותוצג הודעת ניצחון.
- אם נלחץ כפתור ה-w, המשחק נגמר ותוצג הודעת ניצחון.

בכל מקרה של סיום המשחק (בין אם בניצחון במשחק, בלחיצה על w או בהפסד), תוצג הודעה המאפשרת לאפס את המשחק "You Won/Lost! Play again?".
אם נלחץ yes, המשחק יתחיל מחדש.
אם נלחץ no, נסיים את התוכנית.
לבסוף תיצרו את הפונקציה main, אשר מאתחלת את אובייקט BrickerGameManager, וקוראת למתודה run.

אופן ההגשה

1. הגישו את התרגיל כקובץ jar בשם ex3_1.jar.
2. בתוך קובץ זה, ימצאו הקבצים הבאים:
 - a. הספרייה src, המכילה בדיוק את הקבצים המכילים את המימושים שלכם למחלקות המפורטות לעיל ב-API. בכל קובץ תופיע רק מחלקה אחת ששמה כשם הקובץ.
 - b. קובץ בשם README בצורה הבאה:
 1. שורה ראשונה: שם משתמש
 2. שורה שנייה מספר ת"ז
3. אין להגיש את קבצי ה-assets.

בהצלחה!