

תרגיל בית מספר 4

נושא: סימולציית שולחן הוקי אוויר II

דדליין: יום שבת, 5/6/2021, 23:59

הגשה ביחידים

בהצלחה רבה!

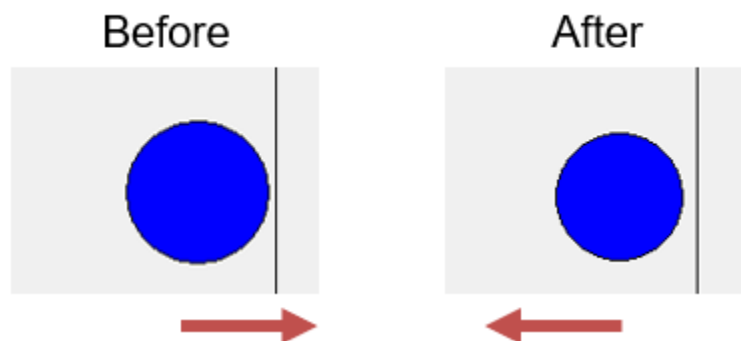
תיאור התרגיל

תרגיל זה מהווה הרחבה של תרגיל מספר 3 בו מימשתם עולם מונחה עצמים של שולחן הוקי אוויר. בתרגיל זה תוסיפו לסימולציה שני סוגי דיסק חדשים. מטרת התרגיל הינה איפיון של יחסי ירושה והתנהגות פולימורפית של אובייקטים, וכן של התמודדות עם יצירה והשמדה של אובייקטים במהלך הסימולציה.

סוגי הדיסקיות בסימולצייה

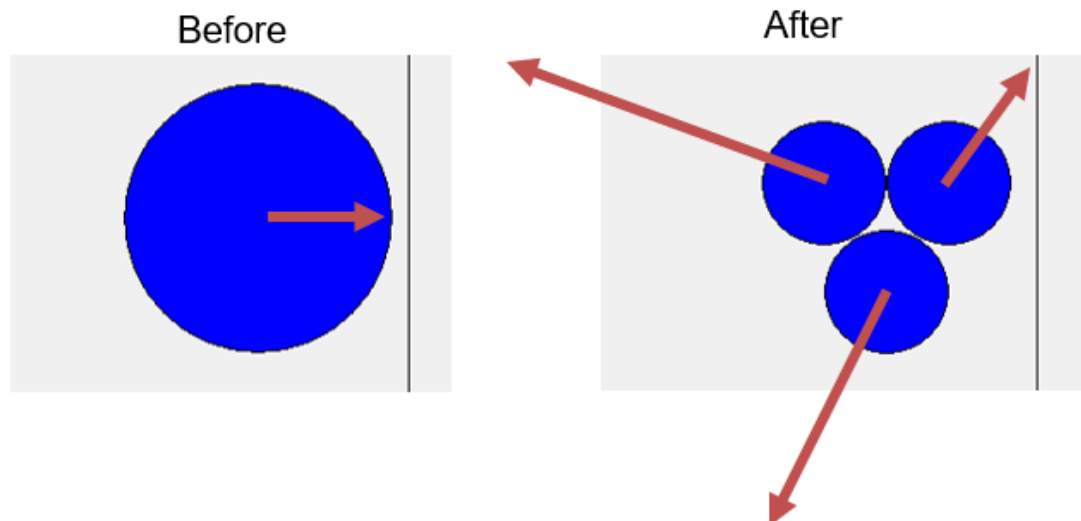
בנוסף לסוג הדיסק הבסיסי שהגדרנו בתרגיל הקודם נוספים שני סוגים חדשים:

1. דיסק מתכווץ – זה דיסק אשר לאחר כל התנגשות רדיוסו מתכווץ בפקטור של 0.9, ונשמד לאחר 3 התנגשויות.



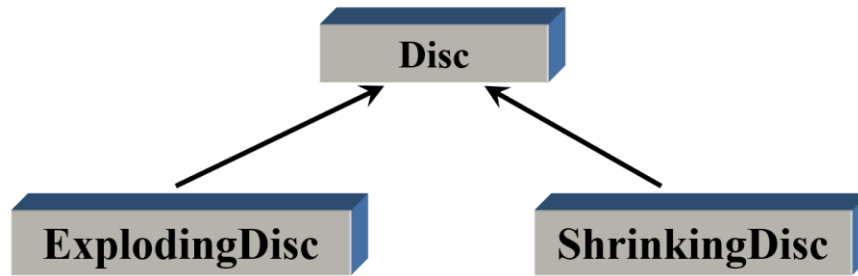
ציור הדיסקה לפני ואחרי ההתנגשות בקיר. אותו כלל חל גם בהתנגשות עם דיסקית אחרת מכל סוג.

2. דיסק מתפוצץ – זה דיסק אשר לאחר כל התנגשות הופך ל-3 דיסקים (כל אחד הינו דיסק מתפוצץ בפני עצמו). גם כאן, דיסק נשמד לאחר 3 התנגשויות. כלומר הדיסק המקורי מדור אפס מתפצל ל-3 דיסקים מדור ראשון כתוצאה מהתנגשות. דיסק מדור 1 מתפצל ל-3 דיסקים מדור 2 כתוצאה מהתנגשות. כאשר דיסק מדור שני מתנגש, הוא נשמד.



תכנות בשפת C++, אביב 2021

שני סוגי הדיסקיות חדשות הינן קודם כל דיסקיות רגילות, עם כל ההתנהגות המוגדרת של תנועה במהירות קבועה והתנגשויות, עם מספר שינויים ותוספות. כלומר, יש לנו כאן ירושה כנראה בציור:



התיאור המדויק של סדר עדכון הדיסקיות וההדפסה, וכן התיאור המדויק של שני סוגי הדיסקיות החדשות מתואר בפסקאות הבאות.

עדכון הדיסקיות וסדר ההדפסה

בדומה לתרגיל 3, על מנת לקדם את הזמן ב- Δt יש עבור כל הדיסקיות לפי הסדר (אשר בהתחלה הוא סדר הדיסקיות בקלט), כאשר העדכון של דיסקית d מתבצע לפי השלבים הבאים (ב-bold מסומנים ההבדלים מתרגיל 3):

- יש לעדכן את מיקום הדיסקית לפי החוק $\text{position} += \text{velocity} * \Delta t$
- לעבור על כל הדיסקיות האחרות, ואם יש התנגשות אז לעדכן את המהירות של שתי הדיסקיות, ולסמן לשתי הדיסקיות שעברו התנגשות.
- יש לעבור על כל הקירות, ואם יש התנגשות אז לעדכן את מהירות הדיסקית, ולסמן שהיא עברה התנגשות.
- אם בשלבים 2 או 3 התגלתה התנגשות של d בדיסקית או קיר, אז יש להחזיר אותה למיקומה שלפני ביצוע שלב 1.

לאחר ביצוע 4 השלבים עבור כל הדיסקיות יש לבצע את הפעולות הבאות:

- לאתחל רשימה חדשה של דיסקיות לרשימה ריקה.
- לעבור על כל הדיסקיות ברשימה הישנה, ועבור דיסקית d:
 - אם לא עברה התנגשות, אז להכניסה לסוף הרשימה החדשה.
 - אחרת, אם היא דיסקית רגילה, אז לנקות את הסימון של ההתנגשות ולהכניסה לסוף הרשימה החדשה.
 - אחרת, אם היא דיסקית מתכווצת, אז לנקות את הסימון של ההתנגשות, לעדכן את רדיוס הדיסקה (אבל להשאיר את המסה ללא שינוי), להגדיל באחד את המונה שסופר כמה התנגשויות עברה. אם פחות מ-3 אז להכניסה לסוף הרשימה החדשה.
 - אחרת (היא דיסקית מתפוצצת), אז לנקות את הסימון של ההתנגשות, להגדיל באחד את המונה שסופר כמה התנגשויות עברה, ואם פחות מ-3 אז להכניס 3 דיסקיות לסוף הרשימה החדשה. חישוב 3 הדיסקיות מתואר בהמשך.
- לעדכן את רשימת הדיסקיות להיות הרשימה החדשה.
- להדפיס את שורת הדיסקיות לפלט.

שלושת הילדים של דיסקית מתפוצצת

נניח דיסקית d עם מיקום $p=(x,y)$, מהירות $v=(v_x,v_y)$, רדיוס r ומסה m. אז הפרמטרים של 3 הילדים שלה, d_0, d_1, d_2 יחושבו לפי האלגוריתם הבא:

```
c = 0.99 // To make sure the new discs are not touching
r' = (2*sqrt(3) - 3) * c * r
a = (4 - 2*sqrt(3)) * r
p0 = p + (0, a), v0 = v + (0, 1), r0 = r', m0 = m/3
```

תכנות בשפת ++C, אביב 2021

$$p_1 = p + \left(r - a, -\frac{a}{2}\right), \quad v_1 = v + \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), \quad r_1 = r', \quad m_1 = \frac{m}{3}$$
$$p_2 = p + \left(-(r - a), -\frac{a}{2}\right), \quad v_2 = v + \left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), \quad r_2 = r', \quad m_2 = \frac{m}{3}$$

דרישות מימוש

אתם חופשיים לתכנן ולממש את התכנית כרצונכם, פרט לדרישות הבאות אותן עליכם לקיים במימוש שלכם:

- הדיסקיות צריכות להיות מופעים שונים (instances) של מחלקה Disc או של מחלקות הנגזרות ממנה.
- הקירות צריכים להיות מופעים שונים (instances) של מחלקה Wall
- יש להשתמש במחלקה Vector2D המיצגת טיפוס של ווקטור בדו-מימד שהוגדרה בתרגיל 3.

פורמט הקלט

זהה לפורמט בתרגיל 3 מלבד העובדה של פקודה disc יש שלוש וריאנטים: disc, exploding_disc, ו-shrinking_disc.

disc <x> <y> <vx> <vy> <radius>

exploding_disc <x> <y> <vx> <vy> <radius>

shrinking_disc <x> <y> <vx> <vy> <radius>

wall <x1> <y1> <x2> <y2>

simulate <T> <dT>

פורמט הפלט

פלט הסימולציה כולל שורה לכל נקודת זמן, מתחילת הסימולציה (זמן אפס) וכל פרק זמן dT עד סוף הסימולציה. פורמט השורה הינו כפי שנראה בדוגמא הבאה, כאשר <t> הינו זמן הסימולציה, והמשתנים <x1>, <y1>, <r1> הם המיקום x,y והרדיוס של הדיסקה הראשונה, וכו'. יש לשים לב שסדר הדפסת הדיסקיות הינו דינמי כפי שתואר קודם, וכן שנוסף הרדיוס לפלט.

<t>: (<x1>,<y1>)/<r1> (<x2>,<y2>)/<r2>

ההדפסה של ערכי x,y,r בפלט הינה בפורמט fixed עם setprecision(4).

הודעות שגיאה

במקרה של שגיאה יש להדפיס את אחת מהודעות השגיאה הבאות ל-stderr ולסיים את הריצת התוכנית:

אם שם הפקודות לא חוקי, או שאי אפשר לקרוא את מספר הארגומנטים הנדרש, או שמגיעים לסוף הקובץ בלי פקודת simulate אזי יש להדפיס:

Error: illegal input.

בהוספת דיסקית/קיר יש לבדוק אם יש התנגשות עם דיסק שכבר קיים על הלוח, ואם כן יש להוציא את אחת משתי הודעות השגיאה הבאות בהתאם למיקרה:

Error: disc to wall collision detected in initial configuration.

Error: disc to disc collision detected in initial configuration.

דוגמא – דיסקית מתכווצת

בהינן קובץ הקלט הבא:

```
shrinking_disc 0 0 1 0 1
wall -3 -10 -3 10
```

תכנות בשפת C++, אביב 2021

```
wall -3 10 3 10
wall 3 10 3 -10
wall 3 -10 -3 -10
simulate 11 0.5
```

מוגדר לנו:

- דיסקה מתכווצת ברדיוס 1 אשר מרכזת ב-(0,0) ומהירותה 1 בכוון החיובי של ציר ה-x.
- שני קירות המקבילים לציר ה-y וממוקמים ב- $x=\pm 3$.
- הגדרת זמן סימולציה כולל של 11 שניות עם זמן עדכון כל חצי שנייה.

הפלט המתקבל הינו:

```
0: (0.0000,0.0000)/1.0000
0.5000: (0.5000,0.0000)/1.0000
1.0000: (1.0000,0.0000)/1.0000
1.5000: (1.5000,0.0000)/1.0000
2.0000: (1.5000,0.0000)/0.9000
2.5000: (1.0000,0.0000)/0.9000
3.0000: (0.5000,0.0000)/0.9000
3.5000: (0.0000,0.0000)/0.9000
4.0000: (-0.5000,0.0000)/0.9000
4.5000: (-1.0000,0.0000)/0.9000
5.0000: (-1.5000,0.0000)/0.9000
5.5000: (-2.0000,0.0000)/0.9000
6.0000: (-2.0000,0.0000)/0.8100
6.5000: (-1.5000,0.0000)/0.8100
7.0000: (-1.0000,0.0000)/0.8100
7.5000: (-0.5000,0.0000)/0.8100
8.0000: (0.0000,0.0000)/0.8100
8.5000: (0.5000,0.0000)/0.8100
9.0000: (1.0000,0.0000)/0.8100
9.5000: (1.5000,0.0000)/0.8100
10.0000: (2.0000,0.0000)/0.8100
10.5000:
11.0000:
```

אנו רואים בפלט את הדיסקה הזו ימינה ושמאלה בין שני הקירות, כאשר אחר כל התנגשות רדיוסה קטן בפקטור של 0.9 (המסה נשארת ללא שינוי). לאחר ההתנגשות השלישית, הדיסקה נעלמת.

דוגמא – דיסקית מתפוצצת

בהינן קובץ הקלט הבא:

```
exploding_disc 0 0 0.5 0 1.5
wall -3 -3 -3 3
wall 3 3 3 -3
wall -3 -3 3 -3
wall 3 3 -3 3
simulate 10 0.5
```

מוגדר לנו:

- דיסקה מתפוצצת ברדיוס 1.5 אשר מרכזת ב-(0,0) ומהירותה 1 בכוון החיובי של ציר ה-x.
- סגורה בתוך ריבוע 6 על 6 אשר מרכזו (0,0)
- הגדרת זמן סימולציה כולל של 10 שניות עם זמן עדכון כל חצי שנייה.

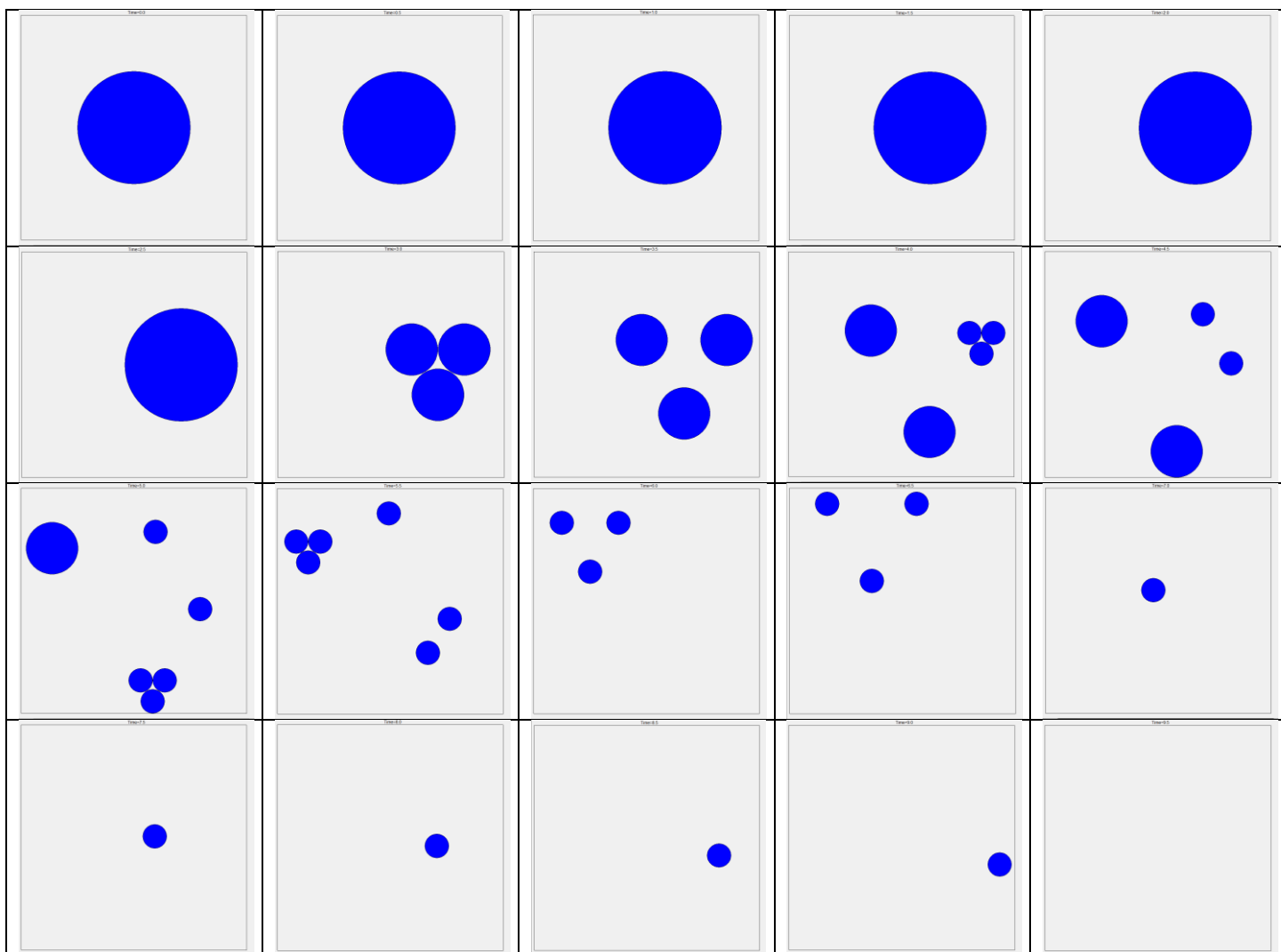
הפלט המתקבל הינו:

```
0: (0.0000,0.0000)/1.5000
```

תכנות בשפת C++, אביב 2021

```
0.5000: (0.2500,0.0000)/1.5000
1.0000: (0.5000,0.0000)/1.5000
1.5000: (0.7500,0.0000)/1.5000
2.0000: (1.0000,0.0000)/1.5000
2.5000: (1.2500,0.0000)/1.5000
3.0000: (1.2500,0.8038)/0.6892 (1.9462,-0.4019)/0.6892 (0.5538,-0.4019)/0.6892
3.5000: (1.0000,1.3038)/0.6892 (2.1292,-0.6519)/0.6892 (-0.1292,-0.6519)/0.6892
4.0000: (0.7500,1.8038)/0.6892 (2.1292,-0.2826)/0.3167 (2.4490,-0.8366)/0.3167 (1.8093,-0.8366)/0.3167 (-0.8122,-0.9019)/0.6892
4.5000: (0.5000,2.3038)/0.6892 (1.9462,-0.0326)/0.3167 (1.1933,-1.3366)/0.3167 (-1.4952,-1.1519)/0.6892
5.0000: (0.5000,2.6732)/0.3167 (0.8199,2.1192)/0.3167 (0.1801,2.1192)/0.3167 (1.7631,0.2174)/0.3167 (0.5773,-1.8366)/0.3167 (-2.1782,-1.4019)/0.6892
5.5000: (1.0029,1.3692)/0.3167 (1.5801,0.4674)/0.3167 (-0.0388,-2.3366)/0.3167 (-2.1782,-1.0326)/0.3167 (-1.8583,-1.5866)/0.3167 (-2.4981,-1.5866)/0.3167
6.0000: (-1.4952,-0.7826)/0.3167 (-0.7423,-2.0866)/0.3167 (-2.2481,-2.0866)/0.3167
6.5000: (-0.8122,-0.5326)/0.3167 (0.3737,-2.5866)/0.3167 (-1.9981,-2.5866)/0.3167
7.0000: (-0.1292,-0.2826)/0.3167
7.5000: (0.5538,-0.0326)/0.3167
8.0000: (1.2369,0.2174)/0.3167
8.5000: (1.9199,0.4674)/0.3167
9.0000: (2.6029,0.7174)/0.3167
9.5000:
10.0000:
```

יותר נוח לתאר זאת בציורים, משמאל לימין ולמעלה למטה:



דגשים

- יש לתכנן מראש את מבנה התכנית, ולהגדיר בהתאם את האובייקטים איתם תעבדו; הקפידו על תכנון מונחה-עצמים של הסימולציה על מרכיביה השונים.
- הקדישו מחשבה למבני הנתונים השונים בהם תעשו שימוש; הביאו בחשבון שיקולי יעילות.
- עבור כל אובייקט יש לבחון מפורשות את הצורך במימוש "שלושת הגדולים" (*Rule of Three*), ולהתייחס בתיעוד למקרים בהם ברירת המחדל מספיקה.

תכנות בשפת C++, אביב 2021

- יש לבדוק תקינות קלטים ולהציג הודעות שגיאה מתאימות. יתכן ובגלל העבודה עם double תהיה בעייה של אי-הסכמה קטנה עם קבצי הרפרנס היותר ארוכים. במקרה כזה, אפשר לבדוק מהי אי ההסכמה באמצעות script פייתון שמצורף או בכל אמצעי אחר שנראה מתאים. הבדקים יודעים על הבעייה, ויערכו בהתאם.
 - בתרגיל בית זה אין להשתמש בספריית STL; מבני הנתונים והאלגוריתמים צריכים להיות ממומשים על-ידיכם.
 - עליכם לוודא כי התכנית עוברת קומפילציית g++ התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
 - עליכם לתעד את הקוד באמצעות הערות קצרות, אך משמעותיות, המתארות את הפונקציות השונות.
 - בייחוד יש להקפיד על מימוש מחלקה של וקטורים דו-מימדיים עם כך האופרטורים הנלווים שמאפשרים עבודה נוחה עימה.
 - יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה:
- התחברו לשרת החוג והריצו hwcheck על הארכיב שלכם, או לחילופין העלו את הארכיב תוך שימוש בפרוטוקול HTML בקישור <https://cs.telhai.ac.il/homework>. לחילופין, התחברו לשרת ה-linux של המכללה, cs.telhai.ac.il והריצו את הפקודה hwcheck עם קובץ ה-zip שלכם כארגומנט.

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('32'), שם התרגיל ('ex4') ותעודת הזהות של הסטודנט/ית המגישה, מופרדים בקו תחתי בפורמט הבא: 32_ex4_studID.zip
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות *.txt *.h *.cpp
- לדוגמא: על סטודנט/ית בעל/ת מספר זיהוי 012345678 להגיש ארכיב בשם 32_ex3_012345678.zip הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.
לא תתקבלנה הגשות באיחור!