

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

מבוא למדעי המחשב 67101

תרגיל 8 - OOP

להגשה בתאריך 04/01/2017 בשעה 22:00

1	מבוא
2	לוח המשחק
2	ספינות
3	תנועת ספינות
4	יירוט ספינות
7	משחק ספינות
8	מימוש
8	Ship
10	Game
13	פיתרון בית ספר
14	הוראות הגשה
14	משחק לדוגמה

מבוא

בתרגיל זה תממשו גרסה של משחק יירוט ספינות. במשחק משחק שחקן המנסה ליירט ספינות הממוקמות על פני לוח דו מימדי. בכל תור רשאי שחקן לירות פצצה למיקום מוגדר על פני הלוח והמחשב בתגובה מדווח לו האם הפצצה פגעה בספינה והאם הספינה הושמדה. בכל תור נעות כל הספינות על פני הלוח, מלבד אלו שנפגעו בתורות קודמים.

תיאור התרגיל מורכב משני חלקים - בחלק המבוא מסופק תיאור מילולי של מהלך המשחק, חוקיו ואופן ניהולו. בחלקו השני של התרגיל ("מימוש") מופיעות הדרישות המפורשות על קטעי הקוד אשר יש לממש והנחיות לגבי בניית המחלקות, הגדרת האינטראקציה ביניהן ודוגמאות לפלטים תקינים של התוכנית.

איך לגשת אל התרגיל – הורידו את קבצי העזר וקבצי השלד (ראו תחת "מימוש"). שימו לב כי פונקציות רבות, בפרט ניהול האינטראקציה עם המשתמש וההדפסות למסך, ממומשות עבורכם בפונקציות הללו. בסוף מסמך זה מוצג משחק לדוגמה וניתנות הוראות לשימוש בפתרון בית ספר, מומלץ בשלב מוקדם להביט במשחק לדוגמה ולהתנסות במשחק כדי להבין איך צריכה להראות התוצאה המוגמרת.

תיאור המשחק

לוח המשחק

- לוח המשחק מורכב ממערכת קואורדינטות דו-ממדית.
- לוח המשחק הוא ריבועי, כלומר אורכו שווה תמיד לרוחבו. לכן, ניתן לזהות לוח עם מידת אורך יחידה המייצגת את אורך צלע הריבוע.
- מערכת המספור של הקואורדינטות מתחילה מהערך - 0.
 - הקואורדינטה $(0, 0)$ היא הקואורדינטה השמאלית עליונה של הלוח.
- כל נקודה (x, y) על גבי הלוח מזוהה על ידי צמד קואורדינטות, הראשונה לציון המימד האופקי (x) והשנייה לציון המימד האנכי (y) .
- כך, בדוגמה שלהלן מוצג לוח משחק מגודל 5 שהקואורדינטה $(1, 3)$ שלו מסומנת בכוכב.

	0	1	2	3	4
0					
1					
2					
3		★			
4					



לוח משחק

ספינות

- ספינה היא אובייקט חד ממדי המונח על לוח המשחק.
- ספינה מאופיינת על ידי:
 - a. אורך - מספר התאים אותם תופסת הספינה על פני הממד בו היא מונחת.
 - b. כיוון (אוריינטציה) - מנח הספינה על פני הלוח. כיוון הספינה יכול להיות מאונך או מאוזן.
 - c. מיקום על פני הלוח - כל ספינה שאורכה גדול מ-1 נמצאת על פני מספר קואורדינטות בלוח. לשם הנוחות נייצג את מיקום הספינה כמיקום הקואורדינטה בעלת הערך המינימאלי מבין מיקומי הספינה.
- כדי לקבוע מה ערך הקואורדינטה המינימאלי (ראו דוגמה בתת הסעיף הבא), נשווה את ערכי הקואורדינטות במימד האוריינטציה של הספינה.
- לדוגמה ספינה בעלת אוריינטציה אופקית אשר גודלה 4 והיא נמצאת בקואורדינטות $[(3,0),(2,0),(1,0),(0,0)]$ הקואורדינטה המינימאלית שלה היא $(0,0)$.


בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- לדוגמה ספינה בעלת אוריינטציה אנכית אשר גודלה 2 והיא נמצאת בקואורדינטות $[(0,3), (0,2)]$ הקואורדינטה המינימאלית שלה היא $(0,2)$.
- d. כך לדוגמה בלוח שלהלן מוצגות שתי ספינות שהקואורדינטה המינימאלית שלהן מסומנות בכוכב.
- הספינה הכתומה היא בעלת אוריינטציה אופקית, אורכה הוא 4, היא נמצאת על התאים $(0,0)$ עד $(3,0)$ על גבי הלוח ולשם נוחות נאמר כי מיקומה הוא $(0,0)$.
- הספינה הירוקה היא בעלת אוריינטציה אנכית, אורכה הוא 2, ומיקומה הוא $(0,2)$.

	0	1	2	3	4
0					
1					
2					
3					
4					

לוח משחק

- הן בשלב אתחול המשחק ובכל שלב במהלך המשחק, אין הגבלה על מספר הספינות השוות באותה קואורדינטה בזמנית.

	0	1	2	3	4
0					
1					
2					
3					
4					

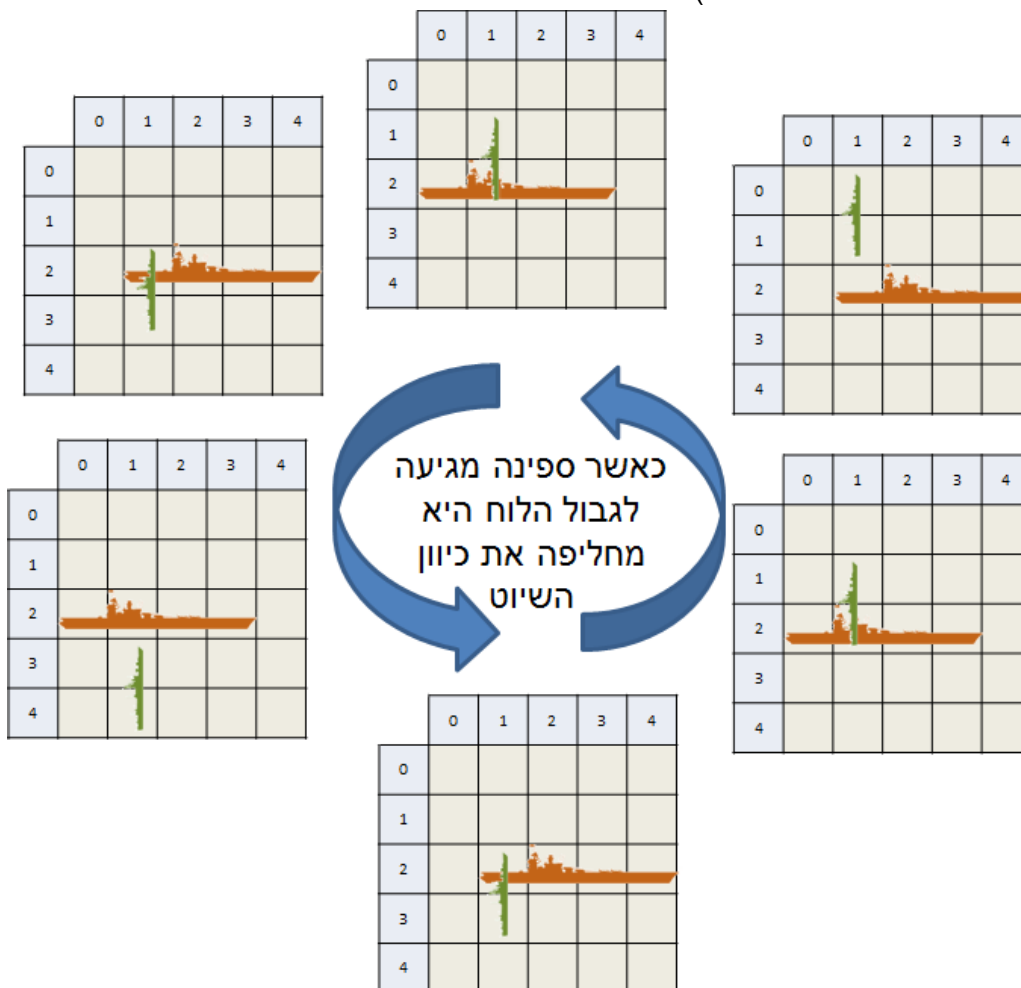
יותר מספינה אחת יכולה להימצא באותה קואורדינטה

תנועת ספינות

- ספינות שטות הלוך ושוב על גבי הלוח במימד האוריינטציה שלהן בלבד. כלומר, ספינה בעלת אוריינטציה מאוזנת שטה ימינה ושמאלה בלבד וספינה בעלת אוריינטציה אנכית שטה למעלה ולמטה בלבד.
- כיוון השייט הראשוני של ספינה מוגדר בעת אתחולה.
- משחק ספינות מתנהל בתורות.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- בכל תור שטה כל אחת מהספינות תא אחד בכיוון השייט שלה.
- כאשר ספינה מגיעה אל גבול הלוח היא מחליפה את כיוון השייט שלה (למעלה-למטה, ימינה-שמאלה).



- החלפת הכיוון מתבצעת בתור הבא ומוצאת מיד אל הפועל, כלומר לא קיים תור בו הספינה לא נעה. סדר פעולות תנועת הספינה בכל תור הוא:
 - אם שייט בכיוון הנוכחי יגרום ליציאה מהלוח:
 - היפוך כיוון התנועה הנוכחי.
 - שייט בכיוון התנועה הנוכחי.
- בפרט:
 - ספינה אשר מאותחלת בגבול הלוח וכיוון התנועה הראשוני שלה הוא לכיוון החוצה מהלוח, תנועתה הראשונה תהיה למעשה בכיוון ההפוך (בכיוון בו היא לא תצא מהלוח).

יירוט ספינות

- בכל תור מנסה שחקן ליירט את הספינות שעל הלוח. יירוט של ספינה הוא בחירה של קואורדינטה על פני הלוח.
- אורך החיים של פצצה:

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- פצצה נשארת על הלוח למשך 3 תורות לכל היותר.
- פצצה מוסרת מהלוח אם :
 - עברו 3 תורות מאז יצירתה (התור בו הונחה על הלוח ועוד שניים). או;
 - הפצצה פגעה בספינה אחת לפחות.
 - ייצוג פצצות על הלוח - במקרה ופצצה לא מתפוצצת, יוצג על הלוח מספר המבטא את מספר התורות שנותרו לפצצה על הלוח, ועוד התור הנוכחי שעבר.
- כלומר פצצה חדשה מיוצגת ע"י 3, לציון שהפצצה יכולה הייתה לפוצץ ספינה בתור הנוכחי (אם הייתה נמצאת בקואורדינטה של ספינה), ונותרו לה עוד 2 תורות כאלה.
- אם הפצצה לא מתפוצצת בכל התורות הבאים : בתור הבא יודפס 2, ובתור השלישי 1.
- בתור שלאחר מכן הפצצה נעלמת ללא פיצוץ, גם אם ספינה נכנסת לקואורדינטה שלה.
- רק פצצה אחת יכולה להימצא בקואורדינטה. אם מניחים פצצה בקואורדינטה בה כבר נמצאת פצצה קודמת הפצצה החדשה מחליפה את הקודמת.
- פגיעה בספינות :
 - פגיעה בספינות מחושבת אחרי שכל הספינות סיימו לנוע.
 - כל הספינות אשר נמצאות בקואורדינטה המיוחסת (חלק כלשהו בספינה, לאו דווקא הקואורדינטה המינימאלית) נחשבות פגועות. אם גוף של יותר מספינה אחת נמצא על קואורדינטה עם פצצה, כל הספינות בקואורדינטה נפגעות.
 - ספינה פגועה לא ממשיכה לשוט, כלומר נשארת במקומה הנוכחי.
 - ספינה שכל תאיה נפגעו נחשבת מושמדת ומוסרת מעל הלוח.
- כך בדוגמה שלהלן :
 - בתור 1:
 - מיירט המשתמש קואורדינטה ריקה לכן בתור הבא נעות הספינות כרגיל.
 - הפצצה ממשיכה להימצא על הלוח גם בתורות הבאים (עד שתתפוצץ או שיסתיימו התורות הנותרים לה על הלוח).
 - בתור 2:
 - מיורטת אחת הקואורדינטות של הספינה האנכית (הירוקה) ועל כן בתורות הבאים היא לא שטה.
 - הספינה האופקית (הכתומה) לעומתה אינה פגועה ולכן ממשיכה לשייט.
 - בתור 3:
 - פוגעת הספינה הסגולה בפצצה הנשארת שם מהתורות הקודמים.
 - כיוון שכל הקואורדינטות של הספינה הסגולה נפגעות בתור זה, היא מוסרת מהלוח בתורות הבאים.
 - מיירט המשתמש קואורדינטה ריקה לכן הפצצה ממשיכה להיוותר על הלוח בתורות הבאים.
 - בתור 4:
 - מיירט המשתמש קואורדינטה בה נמצאות שתי ספינות.
 - הספינה הירוקה פגועה בכל הקואורדינטות שלה ולכן בתור הבא תוסר מהלוח.
 - הספינה הכתומה נפגעת באחת הקואורדינטות שלה ולכן לא תמשיך לשוט בתורות הבאים.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

	0	1	2	3	4
0					
1					
2					
3					
4					

	0	1	2	3	4
0					
1					
2					
3					
4					

	0	1	2	3	4
0					
1					
2					
3					
4					

	0	1	2	3	4
0					
1					
2					
3					
4					

1 תור
משתמש איפה תרצה ליירט? (1, 3)

2 תור
משתמש איפה תרצה ליירט? (0, 0)

3 תור
משתמש איפה תרצה ליירט? (4, 0)

4 תור
משתמש איפה תרצה ליירט? (0, 1)

פצצה על הלוח שתעלם
 ממנו בעוד תור 1/2/3

פגיעת פצצה בספינה
בתור הנוכחי

גוף ספינה אשר נפגע
באחד התורות הקודמים

- כך בדוגמה שלהלן נתון לוח בגודל 4 ושתי ספינות - הירוקה (בגודל 2) והסגולה (בגודל 1).
 - בתור 1:
 - המשתמש מניח פצצה בקואורדינטה (1, 1).
 - הספינות ממשיכות במסלול השיוט הרגיל שלהן.
 - בתור 2:
 - המשתמש מניח פצצה בקואורדינטה (0, 3).
 - הספינות ממשיכות במסלול השיוט הרגיל שלהן.
 - בתור 3:
 - המשתמש מניח פצצה נוספת בקואורדינטה (0, 3). הפצצה החדשה מחליפה את הפצצה הקודמת.
 - הספינות ממשיכות במסלול השיוט הרגיל שלהן.
 - בתור 4:
 - הפצצה שהונחה בתור הראשון בתא (1, 1) עברה 3 תורות ללא פיצוץ ולכן מוסרת מהלוח.
 - הספינה הסגולה עוברת בקואורדינטה (0, 3) שם מונחת פצצה. הפצצה מתפוצצת, פוגעת בספינה ומוסרת מהלוח בתור הבא.
 - הספינה הסגולה נפגעת בכל הקואורדינטות שלה ולכן מוסרת מהלוח בתור הבא.
 - תור 5:
 - המשתמש מיירט את הקואורדינטה (4, 3).
 - במיקומה החדש (לאחר תנועתה) נמצאת הספינה כך שקואורדינטה אחת שלה נמצאת בקואורדינטה שיורטה בתור הנוכחי (4, 3) וקואורדינטה שנייה שלה נמצאת בקואורדינטה של פצצה שהונחה בתור הקודם (4, 4). שתי הקואורדינטות של הספינה נפגעות והיא תוסר מהלוח בתור הבא.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

	0	1	2	3	4
0					
1		3			
2					
3					
4					

תור 1
משתמש איפה
תרצה ליירט?
(1, 1)

	0	1	2	3	4
0					
1		2			
2					
3		3			
4					

תור 2
משתמש איפה
תרצה ליירט?
(0, 3)

	0	1	2	3	4
0					
1		1			
2					
3		3			
4					

תור 3
משתמש איפה
תרצה ליירט?
(0, 3)

	0	1	2	3	4
0					
1					
2					
3					
4					

תור 4
משתמש איפה
תרצה ליירט?
(4, 4)

	0	1	2	3	4
0					
1					
2					
3					
4					

תור 5
משתמש איפה
תרצה ליירט?
(4, 3)

פצצה על הלוח שתעלם
ממנו בעוד תור 1/2/3

פגיעת פצצה בספינה
בתור הנוכחי

גוף ספינה אשר נפגע
באחד התורות הקודמים

משחק ספינות

משחק ספינות מתנהל באופן הבא :

- אתחול המשחק על ידי מספר הספינות הרצוי וגודל הלוח הרצוי.
- יצירת רשימה של ספינות באורך המבוקש:
 - האורך החוקי לכל צוללת בלוח בגודל n הוא בטווח בין 1 ל- $n-1$.
 - בקוד המסופק לתרגיל נתונה הפונקציה `initialize_ship_list` במחלקה `Ship` אשר מייצרת רשימה של ספינות בעלות ערכים רנדומליים. בתרגיל המוגש, יש חובה לקרוא לפונקציה זו מתוך פונקציית האתחול של המחלקה `Game`.
- כדאי ומומלץ בשלב כתיבת התרגיל לשנות את הפונקציה הזו כדי לייצר תרחישים אשר יתאימו לבדיקות נקודתיות של הקוד.
- כל עוד יש ספינות על פני הלוח :
 - בקשת קלט של קואורדינטת יירוט מהמשתמש.
 - תנועת כל הספינות הלא פגועות.
 - עדכון מצב הספינות:
- עבור ספינות נעות - אם במיקומן החדש (לאחר התנועה) נמצאת אחת מהקואורדינטות של ספינה בקואורדינטה בה נמצאת פצצה, קואורדינטה זו נחשבת פגועה.
- לדוגמה אם ספינה בעלת גודל 1, הגיעה בתור הקודם אל הקואורדינטה (0, 1) בתנועה כלפי מטה, בתור הנוכחי המשתמש מבקש ליירט את הקואורדינטה (0, 1), הספינה לא תפגע כי חישוב הפגיעות מחושב לפי מיקומן החדש לאחר התנועה, במקרה זה הקואורדינטה (0, 2).

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- עבור ספינות שאינן נעות (כיוון שנפגעו באחד התורות הקודמים) - אם במיקומן, הזהה לתור הקודם, נמצאת אחת מקואורדינטות גוף הספינה בקואורדינטה עליה תמוקם פצצה, קואורדינטה זו תחשב פגועה.
- קואורדינטה אשר נפגעת פעמיים אינה משנה את מצבה (נשארת פגועה).
- הדפסת הלוח.
- הסרת ספינות מושמדות (שלאחר עדכון מצבן כל הקואורדינטות שלהן פגועות) מהלוח.
- עדכון מצב הפצצות:
- פצצות שהתפוצצו בתור הנוכחי (שפגעו בספינה אחת לפחות) מוסרות מהלוח.
- עבור כל שאר הפצצות עדכון מספר התורות הנותרים להן על הלוח על ידי הפחתה של - 1 ממספר התורות הנותרים בתור הקודם (מספר התורות ההתחלתיים הוא 3).
- דיווח למשתמש על תוצאת היירוט האחרון : כמה ספינות נפגעו וכמה ספינות הושמדו.
- בסיום המשחק יש לדווח למשתמש כי המשחק הסתיים.

מימוש

- בתרגיל ייעשה מימוש בתכנות מונחה עצמים (OOP).
- התרגיל מסופק עם שלד למספר מחלקות.
 - יש לממש את כל הפונקציות של המחלקות, כמפורט להלן.
 - הקבצים המסופקים לתרגיל הם :
- ship.py - מכיל מחלקה המייצגת ספינה בעלת פונקציות אשר יש לממש.
- game.py - מכיל מחלקה המייצגת משחק בעלת פונקציות אשר יש לממש.
- game_helper.py - מכיל פונקציות עזר הממומשות עבורכם.
- ship_helper.py - מכיל פונקציות עזר הממומשות עבורכם.
- ניתן להוסיף מחלקות נוספות (שאתם תיצרו) לצורך פתרון התרגיל.
- ניתן להוסיף פונקציות ומשתנים למחלקות הקיימות אך לא לשנות את חתימות הפונקציות הקיימות.

Ship

מחלקה המייצגת ספינה. הפונקציות הבאות מופיעות בשלד המחלקה ויש לממש אותן :

1. `def __init__(self, pos, length, direction, board_size):`
 - a. פונקציית האתחול של ספינה. מקבלת את:
 - i. מיקום הספינה (מיקום הקואורדינטה המינימאלית), tuple של ערכי x ו y.
 - ii. אורך הספינה.
 - iii. כיוון ההפלגה הראשוני.
 - iv. גודל הלוח בו מוצבת הספינה.
 - v. אפשר להניח כי בבדיקת התרגיל הקלט כולו יהיה תקין ובפרט:
 - 1. length הנו מספר חיובי שלם.
 - 2. pos הוא tuple של מספרים שלמים המייצג ספינה שכולה בגבולות הלוח.
 - 3. direction יקבל את אחד מערכי הקבועים [UP, DOWN, LEFT, RIGHT] מהמחלקה Direction אשר תמומש על ידכם בקובץ Ship.py.
 - 4. board_size הוא מספר שלם וחיובי.
2. `def move(self):`

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- a. פונקציה הגורמת לספינה לנוע יחידת לוח אחת (תא אחד) על גבי הלוח. אם הספינה מגיעה לקצה הלוח, היא משנה את כיוון השייט שלה (ימינה - שמאלה, למעלה - למטה).
- i. ספינה שאחת מהקואורדינטות שלה פגועה לא משנה את מיקומה כלל.
- b. הפונקציה מחזירה אובייקט מהמחלקה Direction (הנמצאת ב - ship.py) המכיל את כיוון התנועה של הספינה.

3. `def hit(self, pos):` :

- a. מודיע לספינה כי פצצה הונחה בקואורדינטה (x, y). הספינה צריכה לעדכן את מצבה הפנימי בהתאמה במקרה של פגיעה באחת הקואורדינטות שלה. אם כל הקואורדינטות שלה נפגעו הספינה מושמדת.
- i. ניתן להניח כי קלט הפונקציה הוא תקין : pos הוא tuple המייצג קואורדינטה חוקית בלוח המורכבת משני מספרים שלמים.
- ii. ערך ההחזרה של הפונקציה מדווח האם הייתה פגיעה חדשה בספינה בתור הנוכחי (True) או לא (False).

4. `def terminated(self):` :

- a. פונקציה הבודקת האם הספינה מושמדת (האם כל הקואורדינטות שלה נפגעו). מחזירה True אם הספינה מושמדת או False אחרת.

5. `def __contains__(self, pos):` :

- a. פונקציה הבודקת האם ספינה נמצאת בקואורדינטה מסוימת. מחזירה True אם אחת הקואורדינטות של הספינה נמצאת בערך ה - (x, y) הנתון או False אחרת.
- i. ניתן להניח כי קלט הפונקציה הוא תקין : pos הוא tuple המייצג קואורדינטה חוקית בלוח המורכבת משני מספרים שלמים.

6. `def coordinates(self):` :

- a. פונקציה המחזירה רשימה של tuples בעל שני ערכים (x, y) המייצגים את כל הקואורדינטות (מיקום אופקי ואנכי בהתאמה) של הספינה בתור הנוכחי. אין חשיבות לסדר האיברים (ה - tuples המייצגים קואורדינטות) ברשימה.

7. `def damaged_cells(self):` :

- a. פונקציה המחזירה רשימה עם מיקומי כל הקואורדינטות (tuples) הפגועות של הספינה (או רשימה ריקה אם אף אחת מהקואורדינטות לא נפגעו).
- b. אין חשיבות לסדר האיברים (סדר הקואורדינטות) ברשימה המוחזרת.
8. `def direction(self):` :
- a. פונקציה המחזירה את כיוון ההפלגה האחרון מתוך המחלקה Direction המוגדרת בקובץ Ship.py. בפרט :
- i. NOT_MOVING אם הספינה נפגעה באחד התורות הקודמים והיא לא נעה יותר.
- ii. אם ספינה הגיעה לקצה הלוח, צריך לדווח על כיוון ההפלגה בתנועה האחרונה ולא על הכיוון בתנועה הבאה שתהיה בכיוון ההפוך.

9. `def cell_status(self, pos):` :

- a. פונקציה המחזירה את מצבה של אחת הקואורדינטות בספינה "פגועה/ לא פגועה". אם הקואורדינטה לא קיימת כחלק מגוף הספינה תחזיר הפונקציה None.

10. `def __repr__(self):` :

- a. פונקציה המחזירה מחרוזת המספקת תיאור של הספינה. אין להשתמש בערכי ההחזרה של פונקציה זו למימוש הלוגיקה של המשחק.
- b. הפונקציה מחזירה המרה למחרוזת של tuple המכיל את הערכים הבאים לפי סדר:

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- i. רשימת כל הקואורדינטות של הספינה.
 - ii. רשימת הקואורדינטות הפגועות של הספינה.
 - iii. כיוון התנועה האחרונה של הספינה.
1. מומרת למחרוזת ע"י הפונקציה `direction_repr_str` הנמצאת בקובץ `ship_helper.py`, המקבלת כקלט את המחלקה `Direction` ואחד הקבועים שלה המייצגים כיוון.
- iv. גודל הלוח עליו נמצאת הספינה.

Game

מחלקה המייצגת משחק ספינות.

1. `def __init__(self, board_size, ships):`
- a. פונקציית האתחול של המחלקה. מקבלת את:
 - i. גודל צלע הלוח - הלוח הוא ריבועי.
 - ii. רשימה של ספינות שיש לכלול במשחק.
 - 1. ניתן להניח כי הקלטים תקינים:
 - a. `board_size` - הנו מספר שלם חיובי.
 - b. `ships` - הינה רשימה של ספינות חוקיות בעלות גודל מתאים (1 עד n-1, עבור n גודל הלוח).
 - c. שימו לב - המשחק צריך לתמוך בכל רשימת ספינות התואמת את דרישות המחלקה `Ship`. - 2. לצורך הרצות (שלכם) במהלך הכנת התרגיל ניתן להשתמש בפונקציות `initialize_ship_list` מהקובץ `game_helper.py` לצורך יצירת רשימה רנדומלית של ספינות.
2. פונקציה המממשת תור יחיד.
- a. אופן המימוש של פונקציה זו נתון לשיקול דעתכם וניתן לממש את הלוגיקה המפורטת להלן במספר פונקציות נפרדות. בבדיקה לא תתבצע אף קריאה מפורשת לפונקציה הזו.
 - b. בתור יחיד מתבצעות הפעולות:
 - i. קבלת קואורדינטה מהמשתמש למיקום הפצצה בתור הנוכחי. - 1. את קבלת הקלט מהמשתמש יש לבצע באמצעות הפונקציה `get_target` מהקובץ `game_helper`.
 - a. בקריאה לפונקציה, הפונקציה מקבלת את גודל הלוח ומבקשת מהמשתמש להכניס את קואורדינטת היעד להנחת פצצה בתור הבא.
 - b. הפונקציה מבקשת מהמשתמש להכניס קלט של שני מספרים המופרדים בפסיק.
 - i. אם מוכנס קלט בפורמט לא חוקי (לא שני מספרים מופרדים בפסיק) הפונקציה תדפיס הודעת שגיאה למשתמש ותמשיך לבקש קלט עד אשר יהיה הקלט חוקי.
 - ii. אם מוכנס קלט בפורמט חוקי של קואורדינטה הנמצאת מחוץ לגבולות הלוח, הפונקציה תדפיס הודעת שגיאה למשתמש ותמשיך לבקש ממנו קלט עד אשר יהיה חוקי.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

c. הפונקציה מחזירה tuple המייצגת את הקואורדינטה המבוקשת להטלת הפצצה.

- ii. השמת פצצה על הלוח בקואורדינטה הנתונה.
 - iii. הזזת כל הספינות שלא נפגעו באחד התורות הקודמים.
 - iv. עדכון פגיעה בספינות על פי הקואורדינטות החדשות שלהן (לאחר התנועה).
 - 1. בפרט - אם הוטלה פצצה על קואורדינטה בה שהתה ספינה בתור הקודם אך בתור הנוכחי בשל תנועתה היא לא נמצאת בה, הספינה לא נפגעת.
 - v. עדכון הפצצות - הפחתת תור אחד מזמן הפגיעה של הפצצות והסרה של פצצות שהתפוצצו בתור הנוכחי.
 - vi. הדפסת הלוח למסך (ראו פונקציה board_to_string).
 - vii. הסרת הספינות המושמדות מהמשחק.
 - viii. דיווח למשתמש על מספר הפגיעות וההשמדות שהיו בתור האחרון.
1. לצורך הדפסת הדיווח למשתמש, יש לעשות שימוש בפונקציה report_turn מהקובץ game_helper.py.

3. `def play(self):`

- a. הפונקציה הראשית אשר מנהלת את המשחק.
 - i. בתחילת המשחק:
 - 1. יש להדפיס למסך את מקרא הסימנים של הלוח (לצורך ההדפסה יש לקרוא לפונקציה report_legend שבקובץ game_helper.py).
 - 2. יש להדפיס למסך את הלוח ההתחלתי.
 - ii. כל עוד יש ספינות על הלוח יש לממש את הלוגיקה לניהול תור המתוארת בסעיף 2 ב - Game ("פונקציה המממשת תור יחיד").
 - iii. לאחר שמושמדות כל הספינות שעל הלוח יש להדפיס למשתמש את ההודעה (לצורך ההדפסה יש לקרוא לפונקציה report_gameover שבקובץ game_helper.py):
'All ships were terminated - Well done!'
- 1. בתור האחרון בו מושמדות כל הספינות הנותרות על הלוח, אין לבצע עדכון נוסף של הפצצות. כך לדוגמה:
 - a. אם בתור הראשון במשחק מונחת פצצה בקואורדינטה (0, 0) אשר לא מפוצצת ספינה אזי יוצג בקואורדינטה (0, 0) המספר 3, לציון שלושה תורות נוספים של הפצצה.
 - b. בתור השני מונחת פצצה על הקואורדינטה (2, 2) אשר מפוצצת ומשמידה את הספינה היחידה שהייתה על הלוח (ספינה בגודל 1). אזי יוצג בקואורדינטה (2, 2) סימן של פיצוץ ובקואורדינטה (0, 0) המספר 2 - כפי שהיה מוצג אילו לא היה מסתיים המשחק (אך ללא עדכון נוסף כלומר לא 1).
 - c. הדפסה של הלוח לאחר סיום המשחק (שלא צריך לבצע אבל אם מישהו היה רוצה לבדוק) צריכה לכלול את הספרה 2 בקואורדינטה (0, 0) בלבד (כיוון והספינה שהייתה שם הושמדה והוסרה מהלוח).

4. `def __repr__(self):`

- a. מחרוזת המספקת ייצוג של לוח המשחק. על הפונקציה להחזיר המרה למחרוזת של tuple המכיל (יש להקפיד על הסדר):
 - i. גודל הלוח.
 - ii. מילון הפצצות הנמצאות על הלוח.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

1. המפתחות במילון צריכים להיות הקואורדינטה בה מונחות הפצצות והערכים במילון בהתאמה צריכים להיות מספר התורות הנותרים לפצצה על גבי הלוח.
- iii. רשימת הספינות הנמצאות על הלוח, לייצוג של כל ספינה יש לעשות שימוש בפונקציה `repr` של המחלקה `Ship`.
- iv. שימו לב כי הפונקציות `str` ו-`repr` המוגדרות כברירת מחדל ב-`Python` על רשימות עושות את הנדרש (אין צורך לכתוב מחדש בעצמכם).
5.

```
def board_to_string(board_length, hits, bombs, hit_ships, ships, debug=False)
```

 - a. פונקציה אשר משמשת להדפסת הלוח.
 - i. הפונקציה ממומשת עבורכם בקובץ `game_helper.py`.
 - ii. הפונקציה מקבלת את קואורדינטות הספינות, הפגיעות והפצצות שעל הלוח ומחזירה מחרוזת המייצגת את המצב הנוכחי של הלוח.
 - b. קלט הפונקציה:
 - i. `board_length` - אורך צלע הלוח.
 - ii. רשימת קואורדינטות מיוחדות לציון על הלוח. בכל ציון של קואורדינטות להלן, הכוונה ל-`tuple` המייצג את ערכי `x` (ערך אופקי) ו-`y` (ערך אנכי) של קואורדינטה. לדוגמה ה-`tuple` בעל הערכים `(0, 4)` מייצג את הקואורדינטה בעלת הערך האופקי `0` והערך האנכי `4`.
 1. `hits` - רשימה של קואורדינטות בהן נפגעה ספינה בתור האחרון.
 2. `bombs` - מילון הממפה בין קואורדינטות למחרוזות. מפתחות (`keys`) המילון הם הקואורדינטות בהן יש כעת פצצות על גבי הלוח וערכי המילון (`values`) הם מספרים המציינים את מספר התורות הנותרים לפצצה על גבי הלוח `(1, 2)` או `(3)`.
 - a. שימו לב - במקרה ונמצאות מספר פצצות על אותה קואורדינטה, יש להדפיס את החדשה ביותר בלבד (זו שנותרו לה יותר תורות על גבי הלוח).
 3. `hit_ships` - רשימה של קואורדינטות בהן נמצא גוף פגוע של ספינה.
 4. `ships` - רשימה של קואורדינטות בהן נמצא גוף לא פגוע של ספינה.
 5. `debug` - האם להדפיס את מיקומי כל הספינות (`True`) או את שורת הפרמטרים בלבד (`False`). ערך ברירת המחדל של המשתנה הוא `False`.- iii. שימו לב - בכל קואורדינטה יכולים להימצא מספר אובייקטים אך יכול להיות מודפס רק ערך אחד. הפונקציה יכולה לקבל קלט כפול, כלומר אותה קואורדינטה תופיע ביותר מרשימה אחת, או באותה רשימה מספר פעמים. במקרה כזה המחרוזת אשר מחזירה הפונקציה תקבע על פי הלוגיקה שלהלן:
 1. אם בקואורדינטה הייתה פגיעה (לפחות אחת) בתור האחרון, הייצוג שלה הוא `'*'`; אחרת
 2. אם בקואורדינטה יש פצצה (זכרו - בכל קואורדינטה קיימת פצצה אחת לכל היותר), הייצוג שלה הוא מספר המייצג את מספר התורות הנותרים לפצצה על הלוח; אחרת
 3. אם בקואורדינטה יש חלק ספינה פגוע (לפחות אחד) הקואורדינטה תיוצג על ידי `'O'`; אחרת

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

4. אם בקואורדינטה יש חלק ספינה לא פגוע (לפחות אחד) הקואורדינטה תיוצג על ידי 'x'.

5. אחרת, כלומר אם לא מתקיים אף אחד מהתנאים שלעיל והלוח ריק בקואורדינטה הזו - הקואורדינטה תיוצג על ידי ' _ '.

שימו לב עיצוב נכון (ומחייב לתרגיל) בסגנון תכנות מונחה עצמים הוא הפרדה בין מימוש של כל הישגיות במשחק ובפרט בין המחלקה Game והמחלקה Ship. המחלקה Game מקבלת רשימה של ספינות בקונסטקטור שלה (בפונקציה __init__) וכל מה ש - Game יכולה להניח על הספינות ברשימה הוא כי הספינות מממשות את הפונקציות הנתונות בספסיפיקציה של המחלקה Ship (הפונקציות move, hit, terminated, coordinates, cell_status, damaged_cells, direction, cell_status וכל הפונקציות הפרטיות כמפורט בתרגיל) ושהן מקבלות קלט ומחזירות פלט כמפורט בתרגיל. אך, אסור ל - Game להניח דבר על המימוש עצמו של הפונקציות הללו. סגנון זה נקרא **design by contract**. מתכנת א' (אתם) אחראי על בניית המחלקה Game ומתכנת ב' (שוב, אתם) אחראי על בניית המחלקה Ship. כל מה שמעניין את מתכנת א' זה שהמחלקה Ship תחזיר פלט תקין עבור קלט תקין (לדוגמה שתחזיר רשימה של tuples המייצגים קואורדינטות חוקיות בלוח כאשר קוראים לפונקציה - coordinates) אבל לא אכפת לו מה הפלט הזה. בפרט, אם מתכנת ב' מחליט להשתגע וקובע שמעכשיו כל פעם שקוראים ל - move הספינה זזה שני תאים קדימה/זזה בכיוון רנדומלי/לא זזה לשום מקום, שהספינות הן בצורת ריבוע או שהן מונחות באלכסון זו בעיה של מתכנת ב' ומתכנת א' לא צריך לדאוג לזה. כל מה שמתכנת א' צריך לדעת הוא מה סוג הקלט שהוא מצופה להעביר כשהוא קורא לפונקציות של Ship ומה סוג הפלט שהוא אמור לקבל.

עיקרון נוסף שיש להקפיד עליו הוא אנקפסולציה. על פי עקרון זה, מחלקות לא חושפות את המשתנים הפרטיים שלהן למחלקות אחרות. במקום, ניהול המשתנים הפרטיים של מחלקה מתבצע מבחוץ באמצעות פונקציות אשר המחלקה בוחרת לחשוף. כך לדוגמה, אם המחלקה Game מכילה רשימה של ספינות הנמצאות כעת על הלוח ואתם מעוניינים להסיר את הספינה השלישית - גישה ישירה אל הרשימה ומחיקת הספינה:

```
del my_game.ship_list[2]
```

תהיה פגיעה בעקרון האנקפסולציה כיוון ורשימת הספינות היא משתנה פרטי של Game ולא צריך לאפשר אליו גישה מבחוץ. במקום, עדיף להגדיר פונקציה פומבית אשר תפקידה להסיר את הספינה מהלוח ולקרוא לה:

```
my_game.remove_ship(ship_index=2)
```

פתרון בית ספר

ממומשת עבורכם גרסה של המשחק. מומלץ לאחר קריאת התרגיל וטרם תחילת המימוש להריץ את המשחק. המשחק אינטואיטיבי ומספר משחקים בו מבהירים את הכוונה הכללית על ניהול המשחק כמו גם שאלות על פרטים ספציפיים.

הפתרון הוא הקובץ המקופמל shipgame הנמצא בתיקייה :

```
~intro2cs/bin/ex8/
```

את פתרון בית הספר ניתן להריץ ממחשבי בית הספר בעזרת הפקודה :

```
~intro2cs/bin/ex8/shipgame <board_size> <ship_number>
```

לדוגמה :

```
~intro2cs/bin/ex8/shipgame 5 2
```

כדי להריץ משחק עם לוח בגודל 5X5 עם שתי ספינות עליו.

הוראות הגשה

יש להגיש קובץ zip יחיד ששמו ex8 המכיל את הקבצים :

1. README

2. קבצי השלד עם המימוש שלכם בתוכם (תזכורת - אפשר להוסיף אך אין לשנות את הקיים בקבצי השלד):

- a. ship.py
- b. game.py

משחק לדוגמה

משחק זה מדגים משחק עם לוח בגודל 5 בו שתי ספינות מונחות על הלוח, הראשונה בגודל 1 והשנייה בגודל 3.

```
*****
LEGEND :
*****
X : Ship body (not hit)
O : Ship body (hit)
* : Hit in last turn
3\2\1 : Life span of bomb
*****
```

```
01234
-----
0|_____
1|_X____
2|_____
3|_XXX__
4|_____
```

Insert x, y coordinate for bomb drop: 0, 0

```
01234
-----
0|3_____
1|X_____
2|_____
3|_XXX__
4|_____
```

In this turn there were 0 hits and 0 terminations

Insert x, y coordinate for bomb drop: 3, 1

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

01234

0|2_____

1|_X_3_

2|_____

3|_XXX_

4|_____

In this turn there were 0 hits and 0 terminations

Insert x, y coordinate for bomb drop: **1, 3**

01234

0|1_____

1|__X2_

2|_____

3|X*X__

4|_____

In this turn there were 1 hits and 0 terminations

Insert x, y coordinate for bomb drop: **4, 4**

01234

0|_____

1|_____*_

2|_____

3|XOX__

4|____3

In this turn there were 1 hits and 1 terminations

Insert x, y coordinate for bomb drop: **4, 4**

01234

0|_____

1|_____

2|_____

3|XOX__

4|____3

In this turn there were 0 hits and 0 terminations

Insert x, y coordinate for bomb drop: **0, 3**

01234

0|_____

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

```
1|_____  
2|_____  
3|*OX____  
4|____2
```

In this turn there were 1 hits and 0 terminations

Insert x, y coordinate for bomb drop: 0, 0

```
01234
```

```
-----
```

```
0|3_____  
1|_____  
2|_____  
3|OOX____  
4|____1
```

In this turn there were 0 hits and 0 terminations

Insert x, y coordinate for bomb drop: 0, 1

```
01234
```

```
-----
```

```
0|2_____  
1|3_____  
2|_____  
3|OOX____  
4|_____
```

In this turn there were 0 hits and 0 terminations

Insert x, y coordinate for bomb drop: 2, 3

```
01234
```

```
-----
```

```
0|1_____  
1|2_____  
2|_____  
3|OO*____  
4|_____
```

In this turn there were 1 hits and 1 terminations

All ships were terminated - Well done!