מגשימים
תכנית הסייבר הלאומית

תרגיל רשות – תכנות מונחה עצמים

Space Wars!!!

נכתב באוניברסיטה העברית בירושלים והותאם לתכנית מגשימים

הקדמה

לאורך אלפי שנים הייתה הגלקסיה מצויה במלחמה מתמדת...
לאחר שמדענית חכמה (שלמדה במגשימים) המציאה עיקרון חדש שנקרא פולימורפיזם, ניתן היה לשדרג את הצי המלכותי ולהוסיף ספינות חדישות עם כישורים מיוחדים. המדענית ביקשה שמישהו או מישהי יעזרו לה לבנות ספינות חדשות ולבדוק את היכולות שלהן.
עזרו לצי המלכותי לסיים את המלחמה, בנו מפעל ספינות קרב בעל יכולות מגוונות.
התרגיל הבא הוא תרגיל לא פשוט, אשר תממשו ב**שפת Java**.
אם המדריך/ה לא ציינה אחרת, את התרגיל ניתן להגיש **ללא דדליין** במהלך סמסטר א'.

המלצות

הגשה: **ניתן לבצע את התרגיל בזוגות**

IDE מומלץ: שימו לב ש Java היא לא שפה שפותחה ב- Microsoft ולכן לא נוח לעבוד עם Visual Studio. ישנן המון פלטפורמות פיתוח מאוד נוחות ל Java. 2 מומלצות יותר מאחרות:

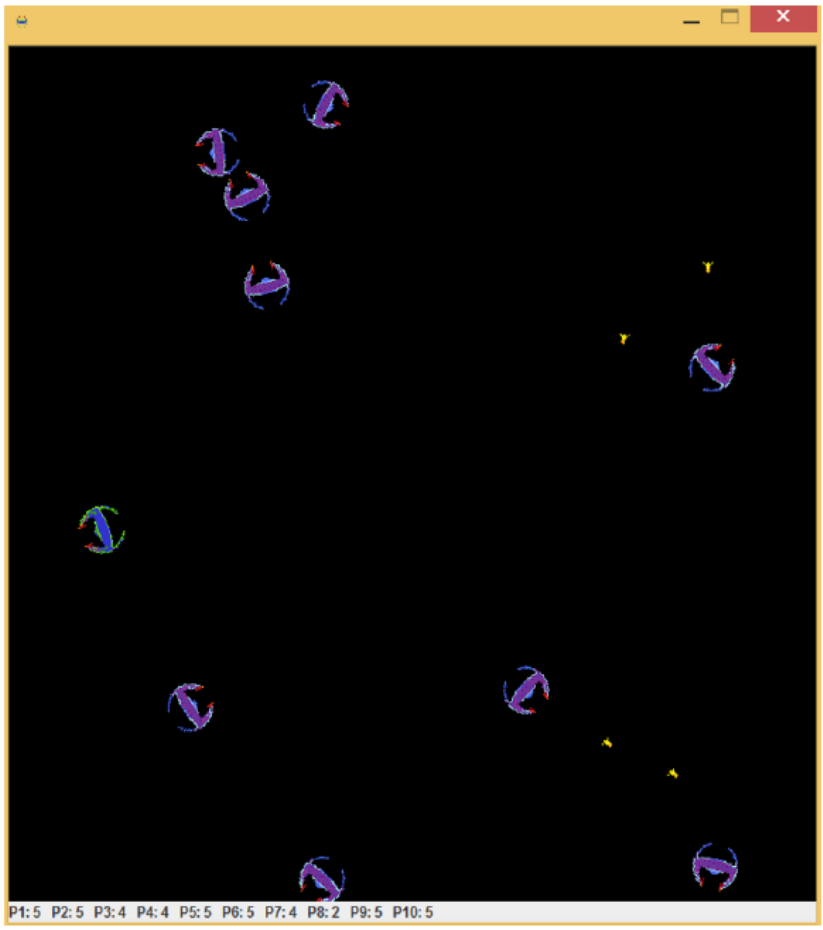
1. IntelliJ של חברת JetBrains (Community). ניתן להוריד ל Windows בקישור https://www.jetbrains.com/idea/download/#section=windows

2. Eclipse של חברת Sun (Oracle). ניתן להוריד ל Windows בקישור https://www.eclipse.org/downloads/download.php?file=/oomph/epp/oxygen/R/eclipse-inst-win64.exe

תיאור כללי

צילום מסך מהמשחק:
בתחתית המסך תוכלו לראות כמה פסילות נשארו לכל חללית

בכל משחק צריך שיהיה **לפחות שחקן אנושי אחד** (צבוע בכחול),
וכמה שחקני מחשב שתבחרו (בוטים).



מסופק לכם/ן קוד הגרפיקה בתוך package בשם oop.ex2, יש לעשות לה import
ולהשתמש בקוד כמו ב"קופסא שחורה" לפי התיאור שתקבלו.

```
import oop.ex2.*;
```

החלליות השונות מיוצגות ע"י אותיות שונות:

h – חללית הנשלטת ע"י אדם (Human controlled).

r – חללית בורחת (Runner).

b – חללית מתנגשת (Basher).

a – חללית תוקפת (aggressive)

d – חללית שיכורה (drunkard)

s – חללית מיוחדת (special)

תוכלו להוסיף חלליות נוספות, כבונוס 😎

המטרה היא לממש את כל החלליות תוך שימוש במנגנון ההורשה ופולימורפיזם.

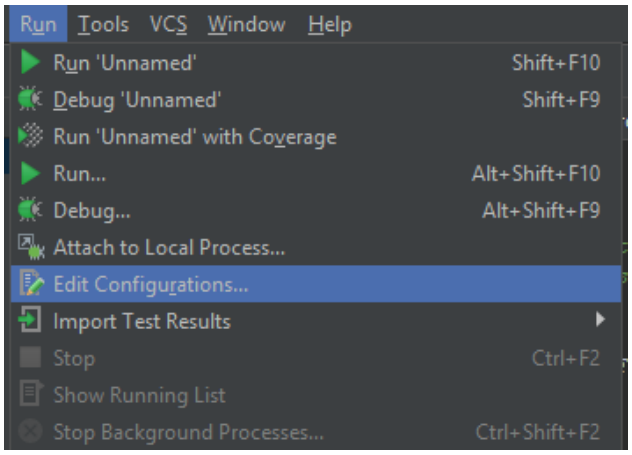
את התרגיל צריך שיהיה אפשר (ב Shell או בתוך ה Editor שלכם) להריץ עם הארגומנטים הבאים:

```
java SpaceWars <space_ship_type> <space_ship_type> ...
```

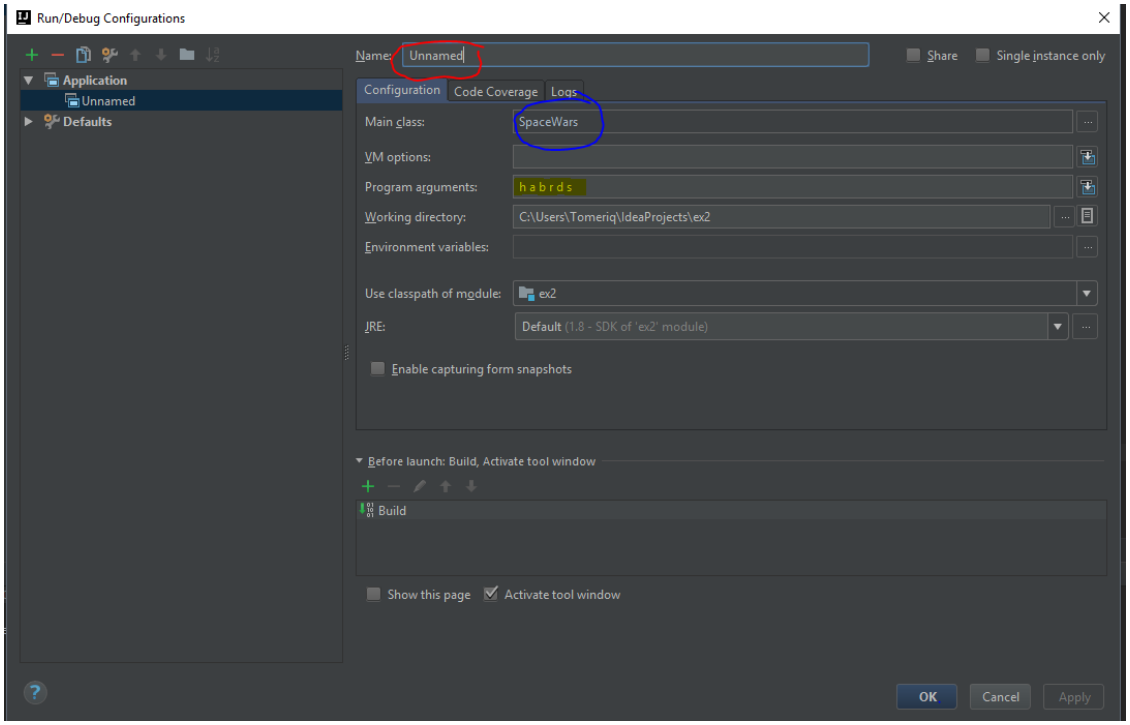
לדוגמא java SpaceWars h a d יריץ משחק עם חללית הנשלטת ע"י אדם, חללית תוקפת, וחללית שיכורה. במידה ואינכם רוצים להריץ את התרגיל ב Shell, ניתן להריץ ב Editor שלכם בצורה הבאה:

לאחר שנפתח פרויקט ונוספה מחלקה עם פונקציית Main, ניתן להריץ את התרגיל ולשלוח ארגומנטים (כמו שעשינו ב C/C++ עם argv, argc).

ניתן להריץ את התרגיל עם ארגומנטים מתוך ה Editor (IntelliJ בדוגמא) כך:



לחצו על לשונית 'run' ואז בחרו 'Edit Configurations'.



לאחר שתגיעו לחלון ההגדרות שימו לב לשלושה דברים:

שם הפרויקט (בדוגמא הוא נקרא Unnamed)

שם המחלקה שבה פונקציית ה Main (צריכה להיות SpaceWars)

הארגומנטים מופרדים ברווח בודד (ויתקבלו כ String ב Main)

חבילת העזר:

מתוך ה package שסופקה לכם/ן (oop.ex2), יש להשתמש בשתי מחלקות בלבד:

SpaceShipPhysics – מייצג את המצב הפיסיקלי של כל חללית (הכיוון, המיקום, הזווית, המהירות וכו') כל חללית צריכה להחזיק מופע בודד של המחלקה הזו. כשהאובייקט נוצר הוא מאותחל בצורה רנדומלית, יש ליצור אובייקט כזה ב3 מקרים בלבד:

א. כשחללית נוצרת
ב. כשחללית מתה ואז קמה לתחייה (אם נשארו לה פסילות)
ג. כשחללית משתגרת

פיתחו את הקובץ SpaceShipPhysics.html בתיקיית API באמצעות דפדפן שתבחרו כדי לראות את ה **API** של המחלקה. (כך תבינו איך להשתמש בה)

GameGUI – אחראית על קלט ופלט גרפי (המנוע).

המחלקה מגדירה 4 תמונות:

- Human Controlled Ship (תמונה של חללית בשליטת אדם)
- Computer Controlled Ship (תמונה של חללית בשליטת מחשב)
- Human Controlled Ship with shield (תמונה של חללית בשליטת אדם עם מגן פועל)
- Computer Controlled Ship with shield (תמונה של חללית בשליטת מחשב עם מגן פועל)

בהמשך תצטרכו שלכל חללית שתממשו תהיה תמונה משלה (ואותה היא תחזיר באמצעות הפונקציה getImage), כדי שתוכלו לעשות זאת תצטרכו לייבא מחלקת עזר באמצעות הפקודה import java.awt.Image .

אם תרצו תוכלו להוסיף תמונות משלכם/ן ולהשתמש בהן ככה שיהיו יותר משני סוגים של תמונות במשחק (תתפרעו! 😇)

בנוסף, באמצעות המחלקה תקלטו מהמשתמש מקשים ותוכלו להתממשק עם החללית שבשליטת אדם.

פיתחו את הקובץ GameGUI.html שבתיקיית API באמצעות דפדפן שתבחרו כדי לראות את ה **API** של המחלקה. (כך תבינו איך להשתמש בה)

קוד נוסף שניתן לכם/ן (וניתן לראות את הAPI שלהן בתיקייה API):

SpaceWars.Java – ה Driver של המשחק, כלומר המחלקה בה נמצאת מתודת ה Main וממנה מתחילה התכנית. **לא לשנות את הקוד במחלקה**. הקוד ניתן לכם/ן, ניתן לנתח ולהבין אותו אבל אין צורך לשנות אותו.

SpaceShip.Java – כרגע יש שם חתימות לא ממומשות של פונקציות שאותם צריך לממש. ניתן להוסיף פונקציות משלכם/ן, והמטרה שתשלימו את הקוד כאהבת נפשכם/ן.

SpaceShipFactory.Java – למחלקה יש בסה"כ מתודה סטטית אחת
(createSpaceShips(String[] שאותה תצטרכו לממש. הפונקציה צריכה לקבל את המחרוזת שמייצגת את החלליות במשחק (לדוגמא h a d s a) ולייצר אובייקטים של מחלקות החלליות הרצויות (בדוגמא, שתי חלליות תוקפות, חללית אדם, חללית שיכורה וחללית מיוחדת).

חוקי המשחק

לכל חללית צריכים להיות המאפיינים הבאים:

- אובייקט מסוג SpaceShipPhysics: שמתאר את המאפיינים הפיסיקליים של החללית
- כמות האנרגיה המקסימלית שיכולה להיות לה
- כמות האנרגיה הנוכחית (מספר בין 0 ל- כמות האנרגיה המקסימלית)
- נקודות חיים (מספר בין 0 ל – 22)

נקודות חיים (Hit Points)

- חללית מתחילה עם 22 נקודות חיים (חיים מלאים)
- כל פגיעה מירייה מפחיתה את נקודות החיים ב1.
- התנגשות עם חללית אחרת מפחיתה את נקודות החיים ב1.
- אם המגן של החללית היה מופעל לא ירדו נקודות חיים (לא מהתנגשות ולא מפגיעה)
- חללית שלא נשארו לה נקודות חיים (0) נחשבת מתה.

אנרגיה (Energy)

-
- כמות האנרגיה המקסימלית מתחילה ב 210, כמות האנרגיה הנוכחית מתחילה ב190
- כאשר חללית מתנגשת בחללית אחרת, והמגן של החללית שהתנגשה היה מופעל, עולה כמות האנרגיה המקסימלית ב - 18, בנוסף גם כמות האנרגיה הנוכחית עולה ב - 18.
  לדוגמא חללית עם מגן דולק שהיו לה 4/190 נקודות אנרגיה, והתנגשה בחללית אחרת תקבל 18 נקודות אנרגיה ויהיו לה 22/208 נקודות.
- כל פגיעה (מירייה של חללית או התנגשות ללא מגן) מורידה 10 מכמות האנרגיה המקסימלית.
- בכל סיבוב עולה כמות האנרגיה הנוכחית ב1 עד לכמות האנרגיה המקסימלית (לא מעבר)
- ירייה (Fire) עולה 19 נקודות אנרגיה

- השתגרות (Teleport) עולה 140
- בכל סיבוב אם המגן של החללית מופעל הוא מוריד 3 נקודות אנרגיה
- כל פעולה שדורשת יותר אנרגיה מכמות האנרגיה הנוכחית לא תתבצע

פעולות החללית

- האצה - החללית תאיץ במעט בכיוון שבו הייתה
- פנייה – החללית תשנה את הזווית שלה ימין או שמאל.

* גם פנייה וגם האצה מתבצעות באובייקט SpaceShipPhysics באמצעות קריאה למתודה ()move. המתודה move צריכה להיקרא **פעם אחת בסיבוב** בכל מקרה (גם אם החללית לא ביצעה שום האצה או פנייה). אם נשלח באותו סיבוב גם פנייה לימין וגם פנייה לשמאל החללית לא תפנה בכלל.

- השתגרות (Teleport) – החללית תיעלם ממיקומה הנוכחי, ותופיע במיקום רנדומלי אחר במשחק. את זה ניתן לעשות ע"י יצירת האובייקט SpaceShipPhysics (שמאותחל רנדומלית) מחדש בתוך החללית.
- ירייה בודדת – ניתן לבצע זאת ע"י קריאה למתודה ()addShot שבתוך מחלקת SpaceWars. אחרי ירייה אחת, **לא ניתן לירות למשך 7 סיבובים**. לדוגמא אם חללית ירתה בסיבוב הראשון, היא תוכל לירות שוב רק בסיבוב השמיני.
- הדלקת מגן – נעשה עבור הסיבוב הנוכחי בלבד.

מכיוון שכמה פעולות שונות יכולות להתבצע בסיבוב אחד, נגדיר את סדר הפעולות כך:

1. Teleport (השתגרות)
2. האצה ופנייה (קורה בו זמנית)
3. הפעלת מגן
4. ירייה
5. התחדשות של נקודה אחת של אנרגיה נוכחית.

כלומר אם נלחצו גם המקש של Teleport וגם מקש תזוזה באותו סיבוב החללית קודם תשתגר ורק אז תעדכן את המיקום.

עדכון של רמת האנרגיה והחיים יתבצע כאשר האובייקט ביצע את כל הפעולות ויידע את המחלקה SpaceWars.

מוות של חללית

כאשר חללית "מתה" SpaceWars יקרא לפונקציה ()reset. החללית תופיע בנקודה רנדומלית עם כמות חיים מקסימלית, וכמות אנרגיה התחלתית (כאילו רק נוצרה).

גם ה cooldown של הירייה מתאפס, כלומר חללית יכולה לירות כשנוצרת שוב גם אם לא חלפו 7 סיבובים.

האובייקט של SpaceWars יעקוב אחרי כמות הפעמים שכל חללית מתה, והמשחק ממשיך עד שלחצו ESC , או עד שסגרו את חלון המשחק.

<u>סוגי חלליות</u>

במשחק תצטרכו לממש כמה סוגי חלליות (כפי שתואר בהקדמה)

1. Human Controlled – חללית בשליטת אדם: 👨👧
   החללית נשלטת ע"י המשתמש באמצעות המקשים:
   'd' ירייה
   's' הדלקת מגן
   'a' השתגרות (Teleport)
   ⬅, ➡ (חץ ימינה/שמאלה) כדי לפנות. ⬆ (חץ למעלה) כדי להאיץ.

2. Runner – חללית בורחת 🏃🏃
   הטייס של החללית הוא פחדן במיוחד, ועושה הכל כדי להימנע מקרב.
   החללית **תמיד תאיץ .**
   אם החללית הקרובה ביותר נמצאת ברדיוס קטן או שווה ל 0.25 יחידות, ואם הזווית של החללית הבורחת ביחס לחללית הקרובה קטנה שווה ל 0.23 רדיינים, החללית הבורחת מרגישה מאוימת ו**תבצע Teleport**.

3. Basher - חללית מתנגשת 🚨🤦
   הטייסת של החללית מנסה להתנגש בכל החלליות האחרות.
   החללית **תמיד תאיץ**.
   אם החללית המתנגשת מגיעה קרוב לחללית אחרת ברדיוס שקטן או שווה ל 0.19 יחידות, היא תנסה **להפעיל מגן.**

4. Aggressive – חללית תוקפת 👺⚔️
   השכנים של הטייס אוהבים לגרור כיסאות, ולכן הטייס של החללית קם מאוד עצבני והוא מנסה לרדוף אחרי כל החלליות האחרות ולירות עליהן.
   החללית **תמיד תאיץ**.
   החללית תמיד תפנה לכיוון של החללית הקרובה ביותר. אם החללית התוקפת מגיעה קרוב לחללית אחרת ברדיוס שקטן או שווה ל 0.19 יחידות, היא תנסה **לירות**.

5. Drankard – חללית שיכורה 🍸🍷🍻

הטייס של החללית בא מכוכב שבו מותר לשתות ולנהוג (אל תנסו בבית!) וכתוצאה מכך החללית טסה בצורה מאוד מוזרה ולא צפויה... ממשו את התנהגות החללית בצורה שתחליטו, בבקשה צרפו לקובץ Readme תיאור של ההתנהגות שלה.

6. Special – חללית מיוחדת 🤖👽

חללית הדגל של המפעל! החליטו איך לממש את החללית המיוחדת שלכם/ן, היא יכולה להיות כזו שתדע להתמודד עם כל סוגי החלליות, או כזו שתעשה דברים מאוד מוזרים במשחק, אפילו יכולה להיות שטותית וחסרת תועלת 😊

Design

ב"מגשימים" לא ראיתם/ן הרבה Java אבל אתם/ן יודעים/ות את העקרונות הנכונים כדי לממש תוכנית מסוג כזה. נסו לכתוב כמה שפחות קוד: השתמשו בהורשה עם מחלקות אבסטרקטיות ופולימורפיזם כדי להפוך את הקוד לגמיש וקל להרחבה.

הערות נוספות למימוש

- כשמופעל המגן הוא מופעל למשך הסיבוב הנוכחי, ובתום הסיבוב אם הכפתור לא נשאר לחוץ המגן יהיה כבוי.
- אין צורך לבדוק את תקינות הקלט של הפונקציות המסופקות לכם/ן.
- מותר להשתמש בספריית Java.Math ובכל המתודות שלה.
- מותר להשתמש בprotected כדי לחלוק משתנים בין מחלקת בסיס ליורשות.
- אסור להשתמש ב Java Reflections. אם הייתם צריכים להשתמש בו כנראה שלא תכננתם נכון את המחלקות.
- ניתן להשתמש בRandom של Java.
- ניתן להוסיף קבצים משלכם/ן (תמונות, טקסט וכו...)
- לא לשנות את הקובץ SpaceWars

Readme

צרפו לפתרון קובץ Readme שבו כתוב איך מימשתם/ן את החללית השיכורה והמיוחדת.
כל דבר נוסף שתרצו להוסיף יכול להירשם שם.

הוראות הגשה

בסוף העבודה הגישו את הקבצים הבאים:

SpaceShip.Java

SpaceShipFactory.Java

כל קובץ אחר שרשמתם/ן (מחלקות נוספות, תמונות וכו'...)

Readme (אפשר בword / pdf / text)

בהצלחה!!!!