

תוכנה 1 תרגיל 8

עולם פראי!



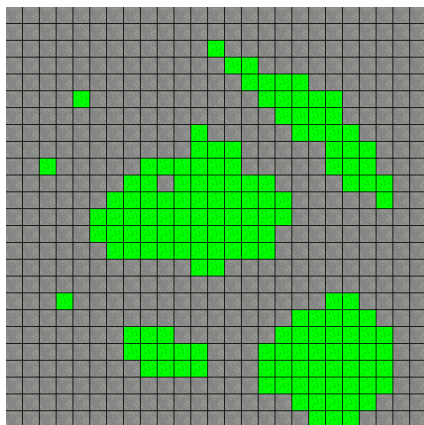
בתרגיל הזה נפתח סימולציה של עולם קטן עם חיות ממינים שונים ועם צמחיה. אתם צריכים לכתוב את תוכנית הסימולציה בעצמכם. התרגיל מתאר את חוקי הסימולציה, מנשקים מסוימים שאתם חייבים להשתמש בהם (כדי שתוכלו לשתף קוד שמגדיר התנהגות של מיני חיות), ומספק רמזים לגבי מבני נתונים אפשריים. אבל את רוב עבודת התכנון והתכנות תעשו בעצמכם.

כחלק מהתרגיל תצטרכו לקבל החלטות מימוש. עליכם לשקול את האופציות העומדות בפניכם ולבחור בזו הראית לכם טובה ביותר. בכל מקום בו נראה לכם שהחלטת המימוש שקבלתם אינה טריוויאלית (שתי אופציות שקולות, אלגוריתם מורכב וכו')

קראו את כל התרגיל וכן את הקבצים המצורפים לפני שאתם מתחילים לכתוב קוד. חשבו כיצד המערכת צריכה לפעול בהינתן הדרישות הפונקציונליות ומגבלות הקוד. הריצו כמה תסריטים על "יבש" (הוסף חיה לעולם, עדכון העולם כתוצאה מפעולה של חיה, ...) כדי להבין טוב יותר את פעולת המערכת לפני כתיבת הקוד.

חוקי הסימולציה הם אלה:

העולם



- העולם הוא עצם מהמחלקה World שמייצגת לוח בגודל n על n של משבצות. בכל רגע נתון, במשבצת יש צמחייה (צבע ירוק באיור) או שהצמחייה נאכלה (אפור).
- העולם הוא מעגלי. את המשבצות שלו נסמן בקואורדינטות $0,0$ (שמאלית תחתונה) עד $n-1, n-1$ (ימנית עליונה). מימין למשבצת $i, n-1$ נמצאת שוב משבצת $i, 0$ ומעל משבצת $n-1, j$ נמצאת משבצת $0, j$. העולם מתנהג באופן דומה בפאות האחרות של הלוח.
- כל משבצת בלוח יכולה להכיל צמחייה וחיה אחת לכל היותר (למעט אירוע זמני שיוסבר בהמשך)

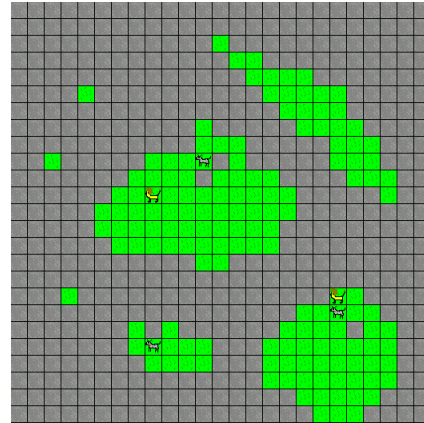
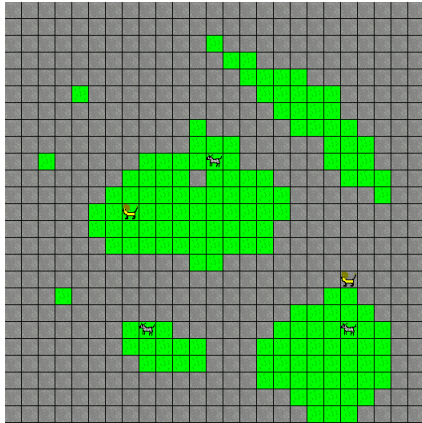
חיות

- חיה היא יצור בעולם שיכול לנוע לאכול ולהשריץ.
- לכל חיה יש כמות אנרגיה מסוימת. פעולות שהיא עושה צורכות אנרגיה, ואכילה מספקת לה אנרגיה. כאשר האנרגיה של חיה נגמרת, החיה מתה ויוצאת מהסימולציה.
- כל חיה שייכת למין מסוים. המינים מתחלקים לשני סוגים: אוכלי עשב וטורפים.
- המין קובע שלושה דברים לגבי החיה: האם היא אוכלת עשב או טורפת, מה גודל הצאצאים שלה (כמות האנרגיה שלהם בלידה), וטווח הראיה שלה. חיה עם טווח ראייה r רואה חלק בגודל $2r+1$ על $2r+1$ של העולם $(r \ll n)$, חלק שהיא במרכזו. לגבי כל משבצת בטווח הראיה, החיה רואה האם יש בה צמחיה, האם יש בה חיה ואם כן מאיזה מין.

הסימולציה

- הסימולציה מורכבת ממהלכים. בכל אחד מהמהלכים כל חיה יכולה לפעול פעם אחת בלבד.
- סדר הפעולה של החיות הוא כסדר הוספתן לעולם למעט צאצאים. המיקום של הצאצא בסדר הפעולות הוא מייד לאחר ההורה. כלומר לאחר שההורה משריץ (ובדרך כלל נע), הצאצא פועל מייד אחריו.
- חיה יכולה לנוע משבצת אחת בכל תור. כלומר, בכל תור היא יכולה לנוע לאחת משמונה המשבצות המקיפות אותה (זכרו העולם מעגלי) במגבלות הבאות:
 - חיה אוכלת עשב יכולה לנוע רק למשבצת שאינה מכילה חיה אחרת (אוכלת עשב או טורפת)
 - חיה טורפת יכולה לנוע גם למשבצת שיש בה חיה אוכלת עשב (אבל לא חיה טורפת).
- אם חיה מחליטה להשריץ ויש לה מספיק אנרגיה לשם כך, נוצרת חיה חדשה מאותו מין באותה המשבצת. אם החיה המקורית לא נעה בתור זה היא מתה (מכיוון שהמשבצת יכולה להכיל חיה אחת לכל היותר).
- כאשר מגיע תורה של חיה לנוע, העולם בודק קודם כל האם היא עדיין בחיים. עצם ההישרדות של חיה עד לתורה עולה לה 0.01 יחידות אנרגיה. אם יש לה פחות, היא מתה מייד כאשר מגיע תורה. אם יש לה יותר מכך, העולם קורא לשירות `act` של החיה.
- כאשר מגיע תורה של חיה לפעול, מופעל השירות `act()` של החיה. שירות זה צריך להחזיר עצם שמתאר את ההחלטה של החיה כיצד לפעול. ההחלטה כוללת שני פרמטרים, האם להשריץ (`spawn`), להתפצל לשתי חיות, אחת בגודל של צאצא של המין והשנייה בגודל שנותר לאחר הקצאת האנרגיה לצאצא) והאם לנוע למשבצת סמוכה (ולאיזו מהן).
- הסימולציה מוציאה לפועל את ההחלטה של החיה (כפי שהתקבלה בעצם המוחזר) היא מוודאת כמובן שהחיה לא חורגת מתנאי הסימולציה (טווח תנועה, תזוזה למשבצת ריקה וכדומה)
 - ראשית, במידה והחיה בחרה להשריץ ויש לה מספיק אנרגיה לשם כך תיווצר חיה חדשה מאותו מין ובעלת אנרגיה התחלתית מתאימה באותה המשבצת של ההורה. האנרגיה הראשונית של הוולד תופחת ממאזן האנרגיה של ההורה.
 - כעת תתבצע התנועה של החיה (באם בחרה לנוע) ועלות התנועה תופחת ממאזן האנרגיה שלה.
- תנועה עולה לחיה 0.05 יחידות אנרגיה
- חיה משיגה אנרגיה ע"י אכילה
 - כאשר חיה אוכלת עשב נעה למשבצת שיש בה צמחייה היא אוכלת את הצמחייה באותה המשבצת (באופן אוטומטי) ומוסיפה למאזן האנרגיה שלה 0.25 יחידות אנרגיה
- כאשר חיה טורפת נעה למשבצת שיש בה חיה אוכלת עשב או כאשר אוכלת עשב נעה למשבצת שבה חיה טורפת החיה הטורפת אוכלת את אוכלת העשב (באופן אוטומטי) ומוסיפה למאזן האנרגיה שלה 0.5 יחידות אנרגיה. החיה הנטרפת כמובן מתה.
- אם חיה אוכלת עשב אכלה את הצמחייה במשבצת מסוימת, המשבצת תישאר ללא עשב לפחות עד לתחילת הסיבוב הבא (לאחר שכל החיות יסיימו את פעולתן)

- בתחילת סיבוב, לפני שהחיה הראשונה בסבב פעלה, עשב צומח מחדש בהסתברות של 25% בכל משבצת קרחת. (יש לבצע את החישוב עבור כל משבצת קרחת בנפרד)
- החיות הראשונות (בתחילת המשחק) נוצרות עם 0.5 יחידות אנרגיה.
- הסימולציה מסתיימת כאשר כל החיות נכחדות, או לאחר 1000 סיבובים.
- שני האיורים הבאים (משמאל לימין) מדגימים מהלך שלם שבו כל החיות פעלו:



מחלקות ומנשקים

נתונים מספר מנשקים ומחלקות שעליכם לממש. תוכלו להוריד את הקבצים מאתר הקורס

• המנשק Tile

המנשק מתאר משבצת אחת בעולם. המנשק מאפשר קריאה בלבד (read only) של מצב המשבצת.

```
/**
 * A single tile in the world.
 * A tile may be occupied by a single animal at a time, and may
 * contain vegetation.
 */
public interface Tile {

    /**
     * Get the animal occupying this tile (if any)
     * @return The animal occupying this tile, or null if none
     * exists.
     */
    public Animal getAnimal() {
        // TODO your code goes here
    }

    /**
     * Check if there is vegetation on this tile.
     * @return true if the tile has vegetation, false otherwise.
     */
    public boolean hasVegetation() {
        // TODO your code goes here
    }
}
```

• המנשק Species

המנשק Species מתאר תכונות קבועות המשותפות לכל החיות מאותו מין. לדוגמא, האם החיה אוכלת עשב או טורפת, מה טווח הראיה שלה ומה היא כמות האנרגיה הראשונית שלה (גודל הצאצא). בנוסף מוגדרת פונקציה היוצרת חיה חדשה ממין זה.

```
/**
 * Represents a specific species (e.g. CowSpecies, LionSpecies).
 * There should only be a single instance of each class implementing
 * this interface.
 */
public interface Species {

    /**
     * Create a new animal that belongs to this species.
     * There should be no other way to create an animal other than
     * this method.
     * @return A new animal object. The return value is never null.
     */
    public Animal newAnimal();

    /**
     * Retrieve the name of the species
     * @return The name of the species (e.g "cow", "lion")
     */
    public String getName();

    /**
     * Determine whether this species is herbivore.
     * A species cannot be both herbivore and carnivore.
     */
    public boolean isHerbivore();

    /**
     * Determine whether this species is carnivore.
     * A species cannot be both herbivore and carnivore.
     */
    public boolean isCarnivore();

    /**
     * Get this species sight range.
     * @return An animal's range of sight.
     */
    public int getRangeOfVision();

    /**
     * Retrieve the size of an offspring. The size of an offspring
     * is the initial energy an animal of this species has.
     * @return a value in the range 0..1
     */
    public double getSizeOfOffspring();
}
```

המנשק Animal

המנשק Animal מתאר חיה בודדת. כל חיה שייכת למין מסוים (מחזיקה הפניה לאובייקט היחיד של אותו מין) ויש לה רמת האנרגיה נוכחית. החיה מאפשרת שינויים חיצוניים ברמת אנרגיה זו אך חל איסור מוחלט על החיה לשנות את רמת האנרגיה בעצמה. בנוסף, לכל חיה יש צוהר דרכו היא רואה את העולם. גודלו של צוהר זה נקבע לפי תווח הראיה של המין. ראו הסבר מפורט למטה.

חיה פועלת בעזרת השיירות `act()`. למעשה בשירות זה החיה מחליטה מה ברצונה לעשות והערך המוחזר של השירות מייצג החלטה זו (ראו למטה). הסימולציה היא שמוציאה לפועל את ההחלטה.

```
public interface Animal {

    /**
     * Plan the animal's next action.
     * @return - An Action object specifying the outcome of
     *         the animal's decision.
     */
    public Action act();

    /**
     * Decrease the animal's energy level by the specified
     * amount
     * @param amount - the amount of energy to decrease. this
     *               * value should be positive
     */
    public void decreaseEnergy(double amount);

    /**
     * get the animal's energy level
     * @return - the animal's current energy level.
     */
    public double getEnergy();

    /**
     * Get the species this animal belongs to.
     * @return - the species object. never null.
     */
    public Species getSpecies();

    /**
     * Increase the animal's energy level by a specified
     * amount
     * @param amount - the amount of energy to increase. this
     *               * value should be positive
     */
    public void increaseEnergy(double amount);
}
```

המחלקה Action

המחלקה Action מתארת את הפעולה אותה מבקשת החיה להוציא לפועל, זוהי התוצאה של שירות ה act. החיה יכולה לציין באם ברצונה להשריץ וכן האם ולהיכן ברצונה לנוע חיה יכולה לנוע משבצת אחת בלבד בכל תור.

```
public class Action {

    /**
     * Specify the horizontal movement of the animal (along
     * the x axis). An animal may move at most a single tile
     * wither way.
     * @return -1, 0 or 1 for left, stay or right respectively.
     */
    public int getHorizontalMove() {
        // TODO Your code goes here
    }

    /**
     * Specify the vertical movement of the animal (along the y
     * axis).
     * @return -1, 0 or 1 for down, stay or up respectively.
     */
    public int getVerticalMove() {
        // TODO Your code goes here
    }

    /**
     * Set the movement for this action. If the movement is
     * illegal a 'stay' (0 on both coordinates) will be set.
     * @param horizontal - movement along the x axis.
     *                 -1, 0 or 1 for left, stay or right respectively
     * @param vertical - movement along the y axis.
     *                 -1, 0 or 1 for down, stay or up respectively
     */
    public void setMove(int horizontal, int vertical) {
        // TODO Your code goes here
    }

    /**
     * Set whether the animal should spawn in this round.
     */
    public void setSpawn(boolean spawn) {
        // TODO Your code goes here
    }

    /**
     * Should the animal spawn in this round?
     */
    public boolean shouldSpawn() {
        // TODO Your code goes here
    }
}
```

המחלקה World

המחלקה World אחראית על מצב העולם ושינויו במהלך הרצת הסימולציה. במחלקה זו תממשו את השירות simulate האחראי על ביצוע הסימולציה. כדי לממש שירות זה תזדקקו להחזיק מידע על מצב העולם והחיות בו. בחרו את מבני הנתונים שלכם בקפידה. לכל חיה קיים צוהר דרכו היא יכולה להסתכל על העולם. חלון זה מוגבל בגודלו בהתאם ליכולת הראיה של החיה. חיה יכולה להתבונן במשבצת בעולם ע"י קריאה לשירות `getTileAt` של העולם. הקואורדינטות הן במונחי החיה, כלומר משבצת (0,0) היא המשבצת שבה היא נמצאת. המשבצת לשמאלה היא (-1, 0) וכו'. העולם יחזיר את ה `Tile` המתאים בהתאם למיקומה של החיה בעולם.

```
public class World {

    /**
     * Create a new world of size n. If the world has already
     * been created this function return null.
     */
    public static World createWorld(int n) {
        // TODO Your code goes here
    }

    /**
     * Get the already created world.
     */
    public static World getWorld() {
        // TODO Your code goes here
    }

    /**
     * Run a simulation of the world.
     * See more details in the assignment description
     */
    public void simulate() {
        // TODO your code goes here
    }

    /**
     * Create a new animal of the specified species at (x, y)
     */
    public boolean createAnimalAt(Species s, int x, int y) {
        // TODO your code goes here
    }

    /**
     * Retrieve the tile at coordinates (h, v) relative to
     * animal's a location.
     */
    public Tile getTileAt(Animal a, int h, int v) {
        // TODO your code goes here
    }

    // TODO choose the internal representation of the world
}
```

בכל הקשור למחלקות ולמנשקים דלעיל חל **איסור מוחלט!** על שינוי המנשקים ושינוי הממשק הפומבי של המחלקות. אין להוסיף או לגרוע פונקציות, לשנות את שמותיהן, לשנות פרמטרים לממש מנשקים וכדומה.

מותר לכם במחלקות אלו להוסיף מימוש (שדות) ושירותי עזר פרטיים בהתאם למימוש שבחרתם.

המטלה

1. ממשו את המחלקה World, מחלקה שמייצגת אוכל עשב (למשל Cow או Zebra וכדומה), מחלקה שמייצגת טורף (Tiger או Lion וכדומה), ואת המחלקות שמייצגות את המינים שלהם (CowSpecies-I LionSpecies למשל). בעת המימוש הקפידו לא לשכפל קוד ע"י שימוש מושכל בירושה ובמחלקות אבסטרקטיות.
2. השירות simulate של World מבצע סימולציה של העולם. בתחילת כל סיבוב בסימולציה השירות ידפיס הודעה, וכן הודעה לאחר שכל חיה פועלת, למשל:


```
*** Round 93 Starting ***
A cow at 24,24 spawned, moved to 25,24, and ate there
A cow at 24,24 moved to 23,24 and ate there
A lion at 30,31 moved to 30,32
A lion at 90,99 moved to 90,100 and ate a cow
```
3. המטרה של החיות והמינים שלהם היא שהמין עצמו ישרוד זמן ארוך ככל האפשר ועם מספר פרטים גדול ככל האפשר. פתחו אסטרטגיות הישרדות חכמות. אפשר להשתמש באסטרטגיות עם מרכיבים רנדומאליים (ראו [java.util.Random](#) ו [Math.Random](#))
4. ממשו מחלקה בשם Play ששירות ה-main שלה יוצר עולם בגודל 100 על 100 עם שלושים אוכלי עשב (מהמין שיצרתם) במשבצות כלשהן, ועם שני טורפים ב-24,74 ו-74,24, ומסמלצת את העולם. מה קורה (מי שורד הכי הרבה זמן, או לאיזה מין יש יותר חיות בסוף הסימולציה?) שימו לב שלא צריך ליצור ממשק גראפי, רק להדפיס את הפעולות באמצעות פונקציות כדוגמת System.out.println
5. [לא חובה] נסו להחליף את החיות שאתם מימשתם בחיות שפיתחו סטודנטים אחרים, איזו חיה שורדת יותר טוב?

הדרכה

אחד הקשיים בתוכנית הזו הוא למצוא את מבני הנתונים מתאימים עבור המחלקה World. עשו שימוש נרחב כל האפשר במחלקות קיימות (Set, List, Map וכדומה). קראו בעיון את התיעוד של המנשקים והמחלקות הממשות אותם ובחרו במימוש המתאים ביותר לכם. הקפידו לשים לב לשימוש באיטרטור בכל אחת מהמחלקות ובפרט מה קורה לאיטרטור קיים כאשר משנים את מבנה הנתונים שהוא פועל עליו(!).

הגדירו תחילה מהם הנתונים להם אתם זקוקים כדי לממש את הסימולציה (מצב העולם, מיקום החיות בו, סדר הפעולה של חיות וכו'). בדקו אילו נתוני זמינים לכם במחלקות והמנשקים שהוגדרו ולאילו תצטרכו להגדיר מבני נתונים בעצמכם.

מומלץ לממש את העולם תחילה ורק אחר כך, כאשר כבר ברור לכם לאילו נתונים אתם זקוקים לצורך הרצת הסימולציה, את שאר המחלקות. בנוסף מומלץ להתחיל במימוש אלגוריתם פשוט עבור החיות ורק אחרי שוידאתם שהמערכת עובדת כראוי ליצור אלגוריתמים מסובכים יותר.

אלגוריתם פשוט במיוחד הוא זוז משבצת אחת קדימה. אלגוריתם מעט מסובך יותר לדוגמא בדוק אם קיים מזון באחת המשבצות הקרובות, אם כן נוע למשבצת זו, אם לא זוז בצורה רנדומאלית.

הוראות הגשה

- קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה ע"י המערכת VirtualTAU (<http://virtual.tau.ac.il>).
- הגשת התרגיל תתבצע ע"י יצירת קובץ zip בנושא את שם המשתמש. לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer.zip.
קובץ ה zip יכיל:
 - קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז. הזהות שלכם.
 - קבצי ה-java. של התכניות שהתבקשתם לכתוב.
 - קובץ טקסט עם העתק של כל קבצי ה Java.