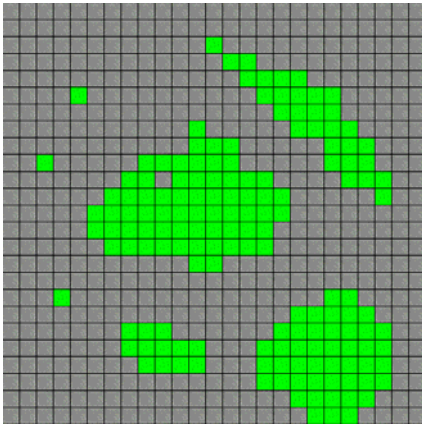


תרגיל 7 בתוכנה 1: עולם פראי!

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.

חוקי הסימולציה הם אלה:



- 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).

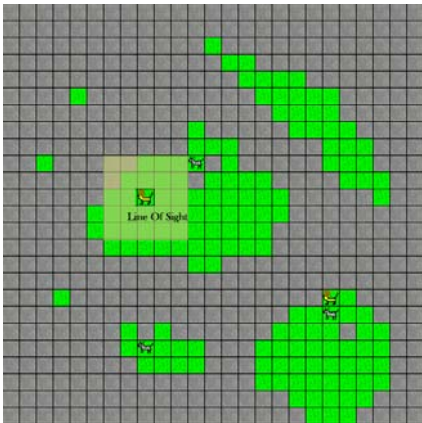
- העולם הוא ציקלי. את המשבצות שלו נסמן בקואורדינטות $0,0$ עד $n-1, n-1$. מימין למשבצת $i, n-1$ נמצאת שוב משבצת $i, 0$ ומעל משבצת $j, n-1$ נמצאת משבצת $i, 0$.

- הסימולציה מורכבת ממהלכים, שבכל אחד מהם כל חיה יכולה לפעול פעם אחת, לפי סדר שיקבע בהמשך. אם חיה אכלה את הצמחייה במשבצת מסוימת, המשבצת תישאר ללא עשב לפחות עד לסוף המהלך. בסוף כל מהלך, עשב צומח באופן אקראי בכל המשבצות הקרחות. ההסתברות שעשב יצמח במשבצת קרחת היא 10%.

- בכל משבצת יכולה גם להיות חיה. משבצת לא יכולה להכיל יותר מחיה אחת.

- לכל חיה יש כמות אנרגיה מסוימת. פעולות שהיא עושה צורכות אנרגיה, ואכילה מספקת לה אנרגיה. כאשר האנרגיה של חיה נגמרת, החיה מתה ויוצאת מהסימולציה. כמות האנרגיה המקסימאלית שחיה יכולה לצבור היא 1.

- כל חיה שייכת למין מסוים, שקובע את ההתנהגות שלה. בסימולציה יש שני סוגי מינים: אוכלי עשב וטורפים.

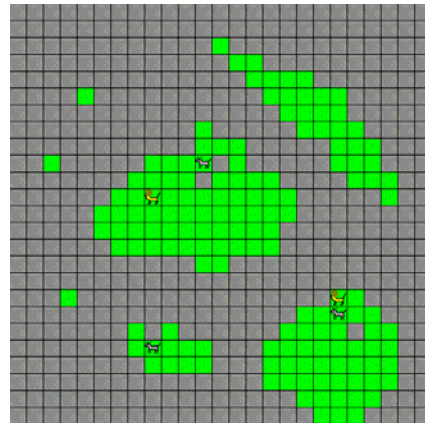
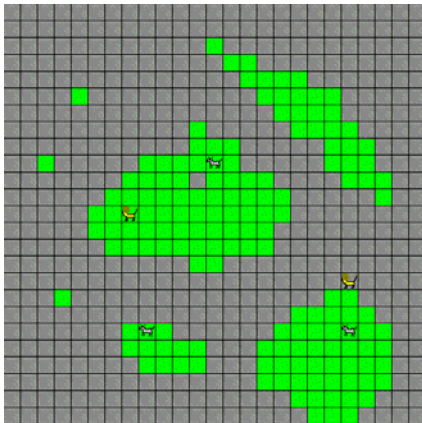


- המין קובע שלושה דברים לגבי החיה: האם היא אוכלת עשב או טורפת, מה גודל הצאצאים שלה (כמות האנרגיה שלהם בלידה), וטווח הראיה שלה. חיה עם טווח ראיה r רואה חלק בגודל $2r+1$ על $2r+1$ של העולם, חלק שהיא במרכזו. לגבי כל משבצת בטווח הראיה, החיה רואה האם יש בו צמחיה או לא, האם יש בו חיה, ואם כן מאיזה מין.

- כאשר מגיע תורה של חיה לפעול, מופעל שירות השירות `act` של החיה. השירות הזה מקבל את כמות האנרגיה שנותרה לחיה, וייצוג של טווח הראיה של החיה, והוא צריך להחזיר עצם שמתאר את ההחלטה של החיה כיצד לפעול. ההחלטה כוללת האם להשריץ

(`spawn`), להתפצל לשתי חיות, אחת בגודל של צאצא של המין והשנייה בגודל שנותר לאחר הקצאת האנרגיה לצאצא) והאם לנוע למשבצת סמוכה. אם החיה מחליטה לנוע, היא קובעת כמובן לאן היא רוצה לנוע. התנועה מוגבלת לשינוי של פלוס או מינוס 1 בכל קואורדינטה (זכרו שהעולם ציקלי).

- חיה אוכלת עשב יכולה לנוע רק למשבצת ריקה. אם במשבצת יש עשב, החיה אוכלת אותו, מקבלת 0.5 יחידות אנרגיה, והעשב נעלם מהמשבצת. הוא אולי יגדל בהמשך, באופן אקראי.
- חיה טורפת יכולה לנוע גם למשבצת שיש בה חיה אוכלת עשב (אבל לא חיה טורפת). בתנועה למשבצת מאוכלסת הטורף אוכל את החיה שבמשבצת, מקבל 0.5 יחידות אנרגיה, והחיה שאכלסה את המשבצת מתה.
- אם חיה מחליטה להשריץ ויש לה מספיק אנרגיה לשם כך, נוצרת חיה חדשה מאותו מין באותה משבצת. אם החיה המקורית לא נעה בתור הזה, היא מתה (מכיוון שהמשבצת יכולה להכיל חיה אחת לכל היותר). המיקום של הצאצא בסדר הפעולות שחיות עושות הוא מייד לאחר ההורה. כלומר לאחר שההורה משריץ (ובדרך כלל נע), הצאצא פועל מייד אחריו.
- כאשר מגיע תורה של חיה לנוע, העולם בודק קודם כל האם היא עדיין חיה. עצם ההישרדות של חיה עד לתורה עולה לה 0.05 יחידות אנרגיה. אם יש לה פחות, היא מתה מייד כאשר מגיע תורה. אם יש לה יותר מכך, העולם קורא לשירות act של החיה. הקריאה עצמה עולה לחיה 0.01 יחידות אנרגיה לכל משבצת שהיא רואה פרט למשבצת שלה עצמה. זה מחיר חוש הראיה של החיה. אם החיה החליטה לנוע ויש לה יותר מ-0.1 יחידות אנרגיה, היא נעה וכמות האנרגיה שלה מופחתת ב-0.01. אם היא גם החליטה להשריץ ויש לה עדיין מספיק אנרגיה לשם כך (האנרגיה של צאצא מאותו מין), היא משריצה (הצאצא נשאר במשבצת המקורית שלה) וכמות האנרגיה שלה מופחתת בהתאם. אם החיה נעה למשבצת שבה היא יכולה לאכול, היא אוכלת (ומקבלת אנרגיה), והאוכל נעלם (עשב או חיה אוכלת עשב שנטרפה). שני האיורים הבאים (משמאל לימין) מדגימים מהלך שלם שבו כל החיות פעלו:



- בסוף המהלך, לאחר שכל החיות פעלו, עשב צומח מחדש בהסתברות של 10% בכל משבצת קרחת.
 - הסימולציה מסתיימת כאשר כל החיות נכחדות, או אחרי 1000 מהלכים.
- הנה מספר מנשקים ומחלקות שאתם צריכים להשתמש בהם, וכן החתימה של השירותים של World.

```
public interface Species {
    public Animal create();
    public String name();
    public boolean herbivore();
    public int rangeOfVision();
    public double sizeOfOffspring();
}
```

```

}
public interface Animal {
    public Species species();
    public Action act(double energyLeft, PatchState[][] view);
}
public class PatchState {
    public Species species;
    public boolean vegetation;
}
public class Action {
    public int xMove; // 0, 1, or -1 only
    public int yMove; // 0, 1, or -1 only
    public boolean spawn;
}
public class World {
    public World(int n) {...}
    // creates an animal at a given location with 0.5 energy
    public void createAt(Species s, int x, int y) {...}
    public void simulate() {...}
}

```

המטלה

1. ממשו את המחלקה World, מחלקה שמייצגת אוכל עשב (למשל Cow או Snail וכדומה), מחלקה שמייצגת טורף (Tiger או Scorpion וכדומה), ואת המחלקות שמייצגות את המינים שלהם (Cows ו-Scorpions, למשל).
2. השירות simulate של World צריך להדפיס הודעה בתחילת כל מהלך, והודעה לאחר שכל חיה פועלת, למשל


```
*** Round 93 Starting ***
A cow at 24,24 spawned, moved to 25,24, and ate the grass there
A cow at 24,24 moved to 23,24 and ate there
A scorpion at 30,31 moved to 30,32
A scorpion at 90,99 moved to 90 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 רנדומליים.
4. ממשו מחלקה PlayTwo ששירות ה-main שלה יוצר עולם בגודל 100 על 100 עם שני אוכלי עשב (מהמין שיצרתם) במשבצות 24,24 ו-74,74, 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?)

5. החליפו את 4 הקבצים הבינריים של החיות שלכם (שניים של חיות ושניים של מינים) עם תלמיד או תלמידה אחרים שפיתחו חיות ממינים עם שמות אחרים משלכם (אחרת זה לא יעבוד). שימו את הקבצים הבינריים של התלמיד/ה האחר/ת בפרוייקט שלכם. יש לציין בפרוייקט עם מי החלפתם חיות.
6. צרו מחלקה PlayFour שמסמלצת עולם בגודל 100 על 100 עם אוכל העשב שלכם ב-24,24, עם הטורף שלכם ב-24,74, עם אוכל העשב האחר ב-74,74, ועם הטורף האחר ב-74,24. מי שורד עכשיו הכי הרבה זמן או עם הכי הרבה חיות? האם לתלמיד/ה האחר/ת יצאו תוצאות דומות עם אותם ארבע מינים?

רעיון למימוש

אחד הקשיים בתוכנית הזו הוא להמציא מבני נתונים מתאימים עבור המחלקה World. סביר להניח שתצטרכו מחלקת עזר שעצמים שלה ייצגו את המצב של חיה מסויימת בעולם (כמות אנרגיה ומיקום). אפשר לייצג את המצב באמצעות שני מבני נתונים שייתחסו לעצמים הללו. מבנה נתונים אחד יאפשר גישה לחיות לפי מיקום בעולם (למשל מערך דו מימדי של התייחסויות) והשני יאפשר גישה לחיות לפי סדר הפעולה שלהן.