Week 7, Lecture 2

Rabbit Population Simulation

Announcements

Assignment 2 is due **4pm** **TODAY**

(late assignments cannot be accepted).

Assignment 2 Submission

- prepare the **three** files

  `Pirate.java`, **`PlankStatistics.java`** **and** **`PercentSaved.java`**

- ensure the top of **each** file contains the comments:

  // Family name: DODGER
  // Given names: Roger The
  // Student number: 87654321

- follow the instructions for submission; in particular, **you must submit all three files together---if you resubmit you must resubmit all three files**
- keep a record of your **Submission Id**

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

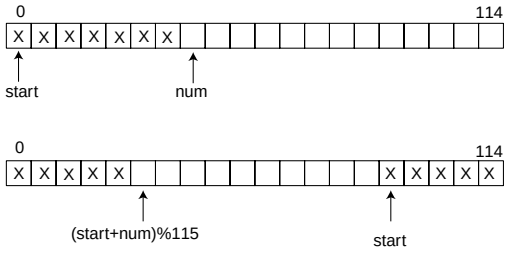
Rabbits live in the above environment, moving freely between the fields. Rabbits have a fixed lifetime but are able to breed. If there are too many rabbits in the environment the available food is consumed and all the rabbits die.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

At each stage each rabbit moves to a vertical, horizontal or diagonal field with equal probability. After a given number of moves a rabbit dies of old age. If at any stage there are 2 or more rabbits in a field a new rabbit is born into that field; the new rabbit moves in the same way and has the same lifetime. If the population of rabbits reaches 100 or more they all die of starvation.

n%4

n/4	0	1	2	3
0				
1	4	5	6	
2	8	9	10	
3	12	13	14	




```
public class Rabbit {

     // instance variables
     private int posn;
     private int age = 0;
     private int lifetime;


     //constructor methods
     public Rabbit (int lt) {
       posn = (int)(16*Math.random( ));
       lifetime = lt;
     }

     public Rabbit (int lt, int n) {
       posn = n;
       lifetime = lt;
     }
```



```
public void move ( ) {
   if (age < lifetime) {
        int next = (int)(16*Math.random( ));
        while (Math.abs(posn%4-next%4)>1
                || Math.abs(posn/4-next/4)>1
                || next == posn)
           next = (int)(16*Math.random( ));
        posn = next;
        age++;
   }
}


public int getPosn ( ) {
   return posn;
}

public int getAge ( ) {
   return age;
}
}
```



```
import genesis.*;

public class Simulation {

  private static int num; // class variables
  private static int lifetime;
  private static Rabbit [ ] rabbits = new Rabbit [115];
  private static int start;

  public static void removeDeadRabbits ( ) {
      for (int i=start; i<start+num; i++) {
          if (rabbits[i%115].getAge( ) == lifetime) {
             start = (start+1)%115;
             num = num-1;
          }
      }
  }
```



```
public static void main (String [ ] args) {
  num = DialogBox.requestInt("number of rabbits:");
  lifetime = DialogBox.requestInt("rabbit lifetime:");
  start = 0;
  for (int i=0; i<num; i++) {
    rabbits[i] = new Rabbit(lifetime);
  }
  RabbitWindow.display(rabbits, 0, num);
  while (num > 0 && num < 100) {
     Delay.milliseconds(1000);
     for (int i=start; i<start+num; i++) {
        rabbits[i%115].move( );
     }
     removeDeadRabbits( );
     mateRabbits( );
     RabbitWindow.display(rabbits, start, num);
  }
}
```



```
 public static void mateRabbits ( ) {
    int [ ] numbersAt = new int [16];
    for (int i=0; i<16; i++) numbersAt[i] = 0;
    for (int i=start; i<start+num; i++) {
       numbersAt[rabbits[i%115].getPosn( )]++;
    }
    for (int i=0; i<16; i++) {
       if (numbersAt[i] > 1) {
         rabbits[(start+num)%115]=new Rabbit(lifetime,i);
         num++;
       }
    }
  }
}
```