CSE 8B Discussion

Wednesday, June 04, 2013

Manuel Espinoza
First of all, Add Listeners!

- Which button(s) will involve the 'running' variable on the listener?
- What should Clear button do?
- How do you "remember" that a button was clicked?
- How do you "remember" which Critter type was clicked?
- Where do you need to add the Critters?
- In which listener should you add the Critters?
Understanding the Critters class.

What do the abstract methods mean? Why do we need to implement those methods on the sub classes and not in the super class? What to do inside each of these methods?
The reactTo method.

public abstract void reactTo( Critter c, Rectangle r );

Different Critters will "react" differently to respect of other Critters.

What are the "reactions" of each Critter?
- Chaser
  - Move to the closest position towards c
- Runner
  - Move to the farthest position from c
- Random
  - Move randomly to any position less than 10 pixels from current location
- Custom
  - Any ideas?
Chaser reactTo

- What are the possible positions that the Chaser can have given the restriction of moving only one pixel?
- Which of these positions is the most appropriate for the Chaser definition?
- How can we get that position?
- How can we get the distance between two critters?
What about the rectangle?

This is used to control the boundaries.

How do you determine if a Critter is out of the boundaries?
The Interactor class.

- It will only need the interact method:

```java
public void interact(Critter c, ArrayList<Critter> cList, Rectangle r)
```

- Get the closest Critter from cList (ignore when it is itself)
- Call reactTo with Critter c and the found closest Critter.
Given the following method signature:

```java
public static void printClosestInt(int x, int[] d)
```

Write the method body so that it will print the closest integer in the array `d` to the value `x`, ignore the value if it is equal to `x`.

i.e. `x = 4 d= {6,8,0,1,0,5,10}` will print 5

```java
public static void printClosestInt(int x, int[] d)
{
    int closest=-1;
    int lastDiff=Integer.MAX_VALUE;
    for(int n: d){
        if(n!=x){
            int diff = Math.abs(x-n);
            if(diff<lastDiff){
                closest = n;
                lastDiff = diff;
            }
        }
    }
    System.out.println("The closes int is: "+closest);
}
```
Questions?

Contact Info: m8espino@eng.ucsd.edu
Feedback

Please submit your evaluations:

http://academicaffairs.ucsd.edu/Modules/Evals/
THE END!