PA9: Critters

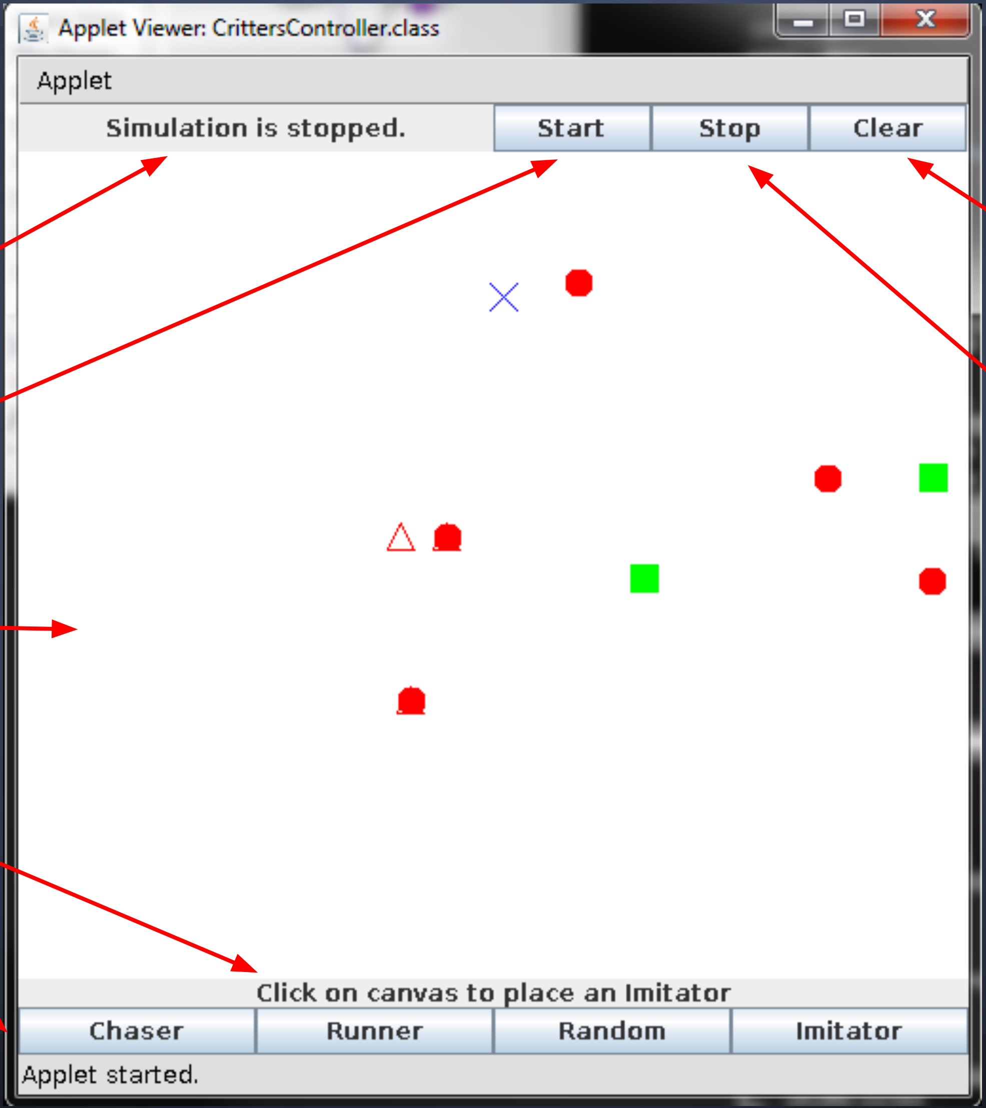
Ren Lee
Katherine Ma
CSE11 / FA12

Required Files

- CrittersController.html
- CrittersController.java
- CrittersSimulator.java
- Critter.java
- Chaser.java
- Runner.java
- Random.java
- README

...and Imitator.java if you do extra credit

Example of what it looks like



Applet Viewer: CrittersController.class

Applet

Simulation is stopped.

Click on canvas to place an Imitator

Applet started.

Simulation state text

Simulation start button

Where critters run around

Critter message

Critter types

Clear button - removes all critters

Simulation stop button

Test Drive time!

Let's see an example...

High-level View

CrittersController - create all the GUI components, layout
                              - create instance of CrittersSimulator

CrittersSimulator - control animation

- -> Critters (abstract class)
    - -> Chaser
    - -> Runner
    - -> Random
    - -> Imitator (extra credit)

CrittersController.java

- Controller Class
- in begin(), create all the buttons, GUI layout, encouraged to use JPanel
- use BorderLayout, GridLayout, etc. as needed
- extend WindowController
- implement ActionListener (listen to the 6 buttons)
- JButton for buttons, JLabel for status messages
- When starting up, should begin in "start" state

CrittersController.java

- Most likely, you'll use these methods
  - begin()
  - onMouseClick()
  - actionPerformed()
  - ...a custom method to change status label using setText() to change text variable

CrittersController.java

- Instance Variables that will help you...

  ArrayList<Critter> critters

  - use to keep track of critters on canvas

  CritterSimulator simulation

  - an instance of CritterSimulator class

CrittersController.java

- Simulation State Message Rules

  When in stopped state, should read "Simulation is stopped"

  When in started state, but with less than 2 Critters on canvas, should read "Please add two or more critters." (i.e when program begins)

  When in started state and there are 2 or more Critters on canvas, should read "Simulation is running."

- Have your custom method deal with this logic, don't spread it out everywhere

CrittersController.java

- Critter selection message rules:

  "Select which critter to place: " when no critter type is selected
  "Click on canvas to place a [critter type]" when a critter type button is clicked

- All critters will be created based on their <u>center point</u>

Example of CrittersController.java

AN EXAMPLE OF HOW YOURS MIGHT LOOK LIKE

```
//import appropriate libraries
public class CrittersController extends WindowController implements ActionListener{
	//variables you might need

	public void begin(){
		//layout GUI here
	}

	public void onMouseClick(Location loc){
		//determine which critter was selected and place it on the canvas
	}
```

Example of CrittersController.java (Cont.)

```
    public void actionPerformed( ActionEvent evt){
        //distinguish which buttons were clicked
        // start/stop/clear
        // chaser/runner/random/imitator
    }


    // You might want to have a private method that changes Simulation status label

    // add whatever private methods you might want to use (i.e: setters/getters)

}
```

CrittersSimulator.java

- Most likely you'll have:

  class should extend ActiveObject

  DELAY variable set to 40~50

  an ArrayList<Critter> variable, which is set by the constructor

  Boolean variable to keep track of whether simulation status (true for running, false for not running), and a setter and getter for this

  CrittersSimulator constructor that takes an ArrayList<Critter> as its parameter so that both Controller and Simulator have reference to same critters Arraylist

  A run() method

CrittersSimulator.java

constructor:

```
public CrittersSimulator(ArrayList<Critter> critters){
    //set boolean var that keeps track of whether simulation is running to true
    this.critters = critters;
    start();      //DON'T FORGET!
}
```

CrittersSimulator.java

- Should only run when there are 2 or more critters on canvas.
- run() method will calculate distance from every Critter to all OTHER critters to determine which one is closest, then call reactTo() method for that closest Critter

Lots of ways to do this...

- matrix
- hardcore looping with for loops
- anything else you can think of

CrittersSimulator.java

```
public void run(){
    while(true){
        pause(DELAY);
        ...
        //all your calculation stuff
        //remember to check: Chasers don't chase Chasers
        //determine which critter to call reactTo()
    }
}
```

- IMPORTANT: do not move pause()
- should not have another while loop inside
  -- LAG!!

CrittersController.html

- Same as .html used for previous assignments, but modified
- Code Snippet:

```
<applet
  code = "ResizableBallController.class"
  ....
>

change to
<applet
  code = "CrittersController.class"
  ...
>
```

Critter.java

- abstract class
- should have variables and methods that are common to all Critter type classes

- Chaser, Runner, Random (and Imitator) should extend Critter

Critter.java

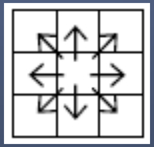
- protected Location pt -- the center of the critter
- protected DrawingCanvas canvas
- other constants you might find useful (SIZE, ...)
- public Critter(Location loc, DrawingCanvas canvas)
  - Constructor <MUST - at least have this>
- public abstract void reactTo(Critter other);
  - must have this
- public abstract void kill();
  - removes critter from canvas
- public Location getLoc()
- public void setLoc(Location loc)

Each critter should have

- its own constructor
  - super()
  - bounds check
- reactTo()
- kill()
      - removeFromCanvas()

public void reactTo(Critter other)

- passed in reference to other Critter
- current Critter can update its location based on location of other Critter (except for Random)
- distanceTo(), Double.MAX_VALUE...will be useful...
- imagine each Critter moves on a grid like this:

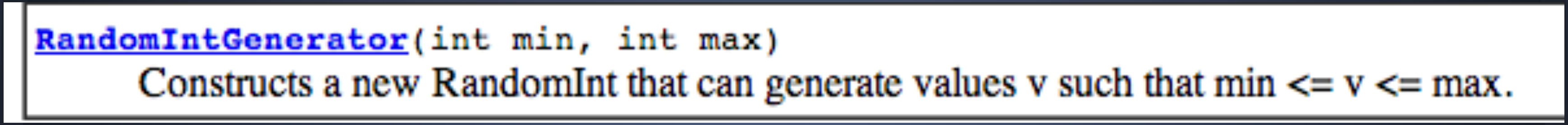


x, y coordinate of Critter will change by -1 or 1 each time

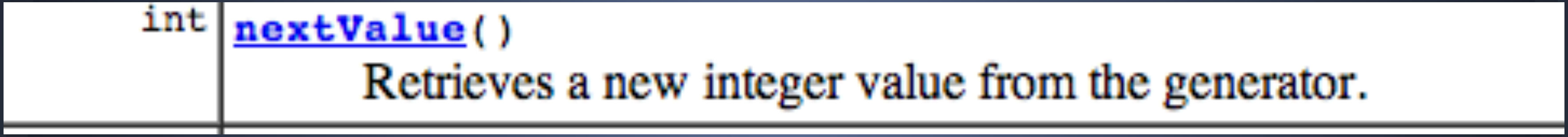
(-1,1)	(0,1)	(1,1)
(-1,0)	CRITTER	(1,0)
(-1,-1)	(0,-1)	(1,-1)

public void reactTo(Critter other)

- Grid movement only applies to Chaser and Runner
- Random's x,y coordinates should change by a random value between [-10,10]
- Random's x and y should be generated <u>separately</u>, not one value for both



`RandomIntGenerator(int min, int max)`
Constructs a new RandomInt that can generate values v such that min <= v <= max.



`int`	`nextValue()` Retrieves a new integer value from the generator.

public void reactTo(Critter other)

```
public void reactTo( Critter other ) {
    if( other == null ) return;        //don't do anything if other is null
    for (all the 8 possible new locations)
        //check distance to this temporary location
        //check if this distance is the max(runner)/min(chaser) distance
        //make sure to check temporary location with the world boundaries!
    //set the new location
    }
```

Example of ctor for a Critter type

For example, for Runner

```
public Runner(Location location, DrawingCanvas canvas){
    super(location, canvas);
    //set the center point X and Y making sure X and Y are within bounds
    rect = new FilledRect(...);
    rect.setColor(...);
    }
```

Chaser.java

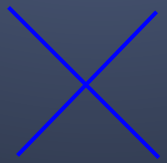
- RED circle FilledOval object
- diameter 15 pixels
- move <u>TOWARDS</u> Critter that is closest
- should never chase after other Chasers
    - can assume won't be another Chaser if dealt with in CrittersSimulator

Runner.java

- GREEN square FilledRect object
- 15x15 pixels
- move AWAY from other closest Critter
- if runner touches any of borders of the canvas, should be moved to a randomly selected location somewhere inside canvas

Random.java

- BLUE X using two individual line objects
- bounding box 15x15 pixels
- move randomly from current location to nearby location
    - RandomIntGenerator
- not influenced by other Critters' movements
- Chasers and Runners will be influenced by Randoms



IMPORTANT:

- No critter can move outside canvas.
  - (Randoms: Careful!)
- This should be determined by each Critter's
  <u>perimeter</u> not center.
- Remember: no magic numbers!
- Can assume canvas size will not change.

Extra Credit

- Imitator.java
- ...and other functionalities

This is up to you.

Don't forget README.

Any questions?

Start Early!
Have fun!

She definitely will ->