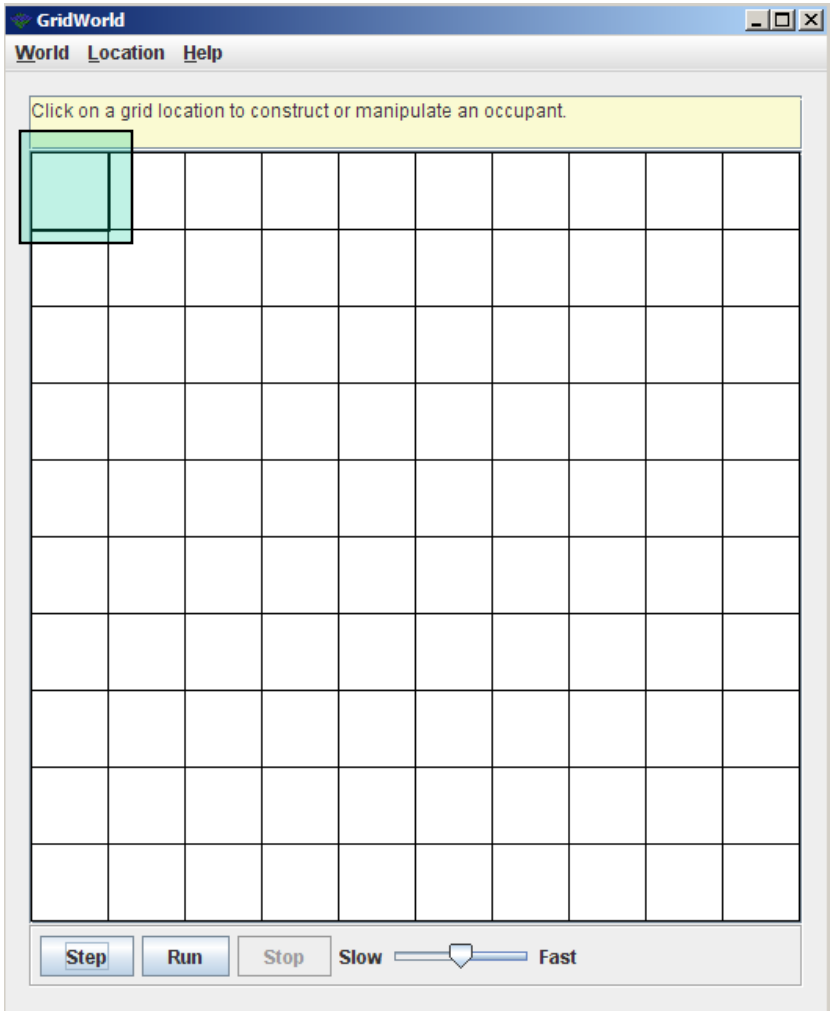
GridWorld

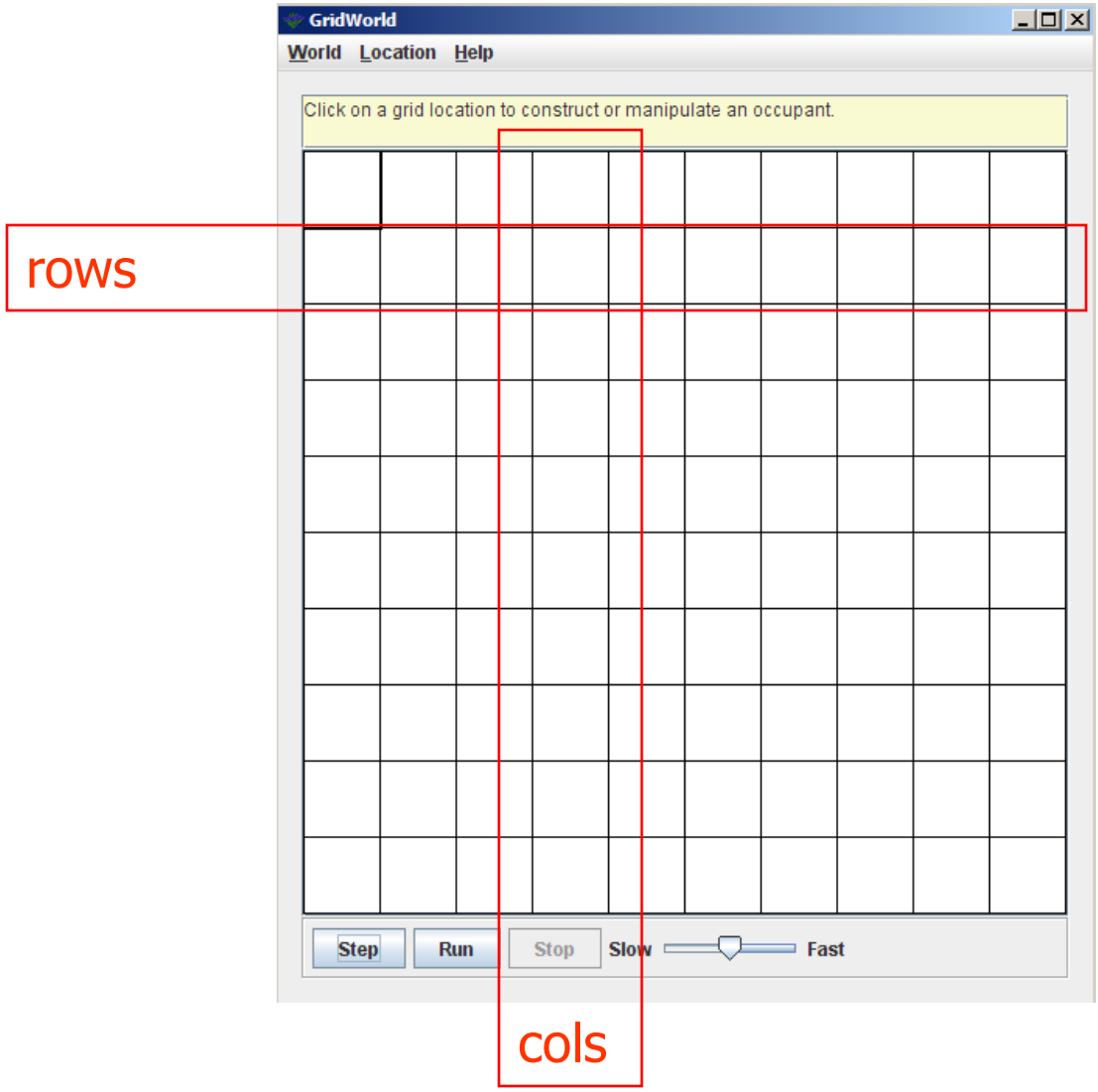
What is GridWorld?



Row = 0

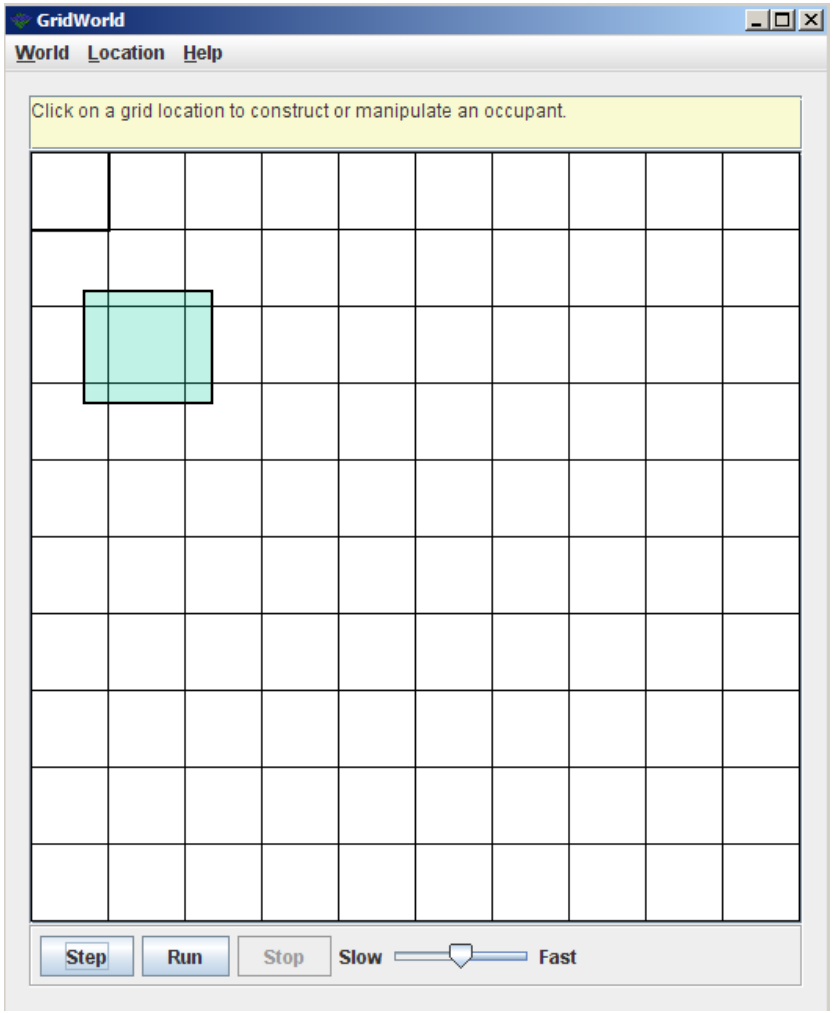
Column = 0

What is GridWorld?



A grid is a structure that has rows and columns.

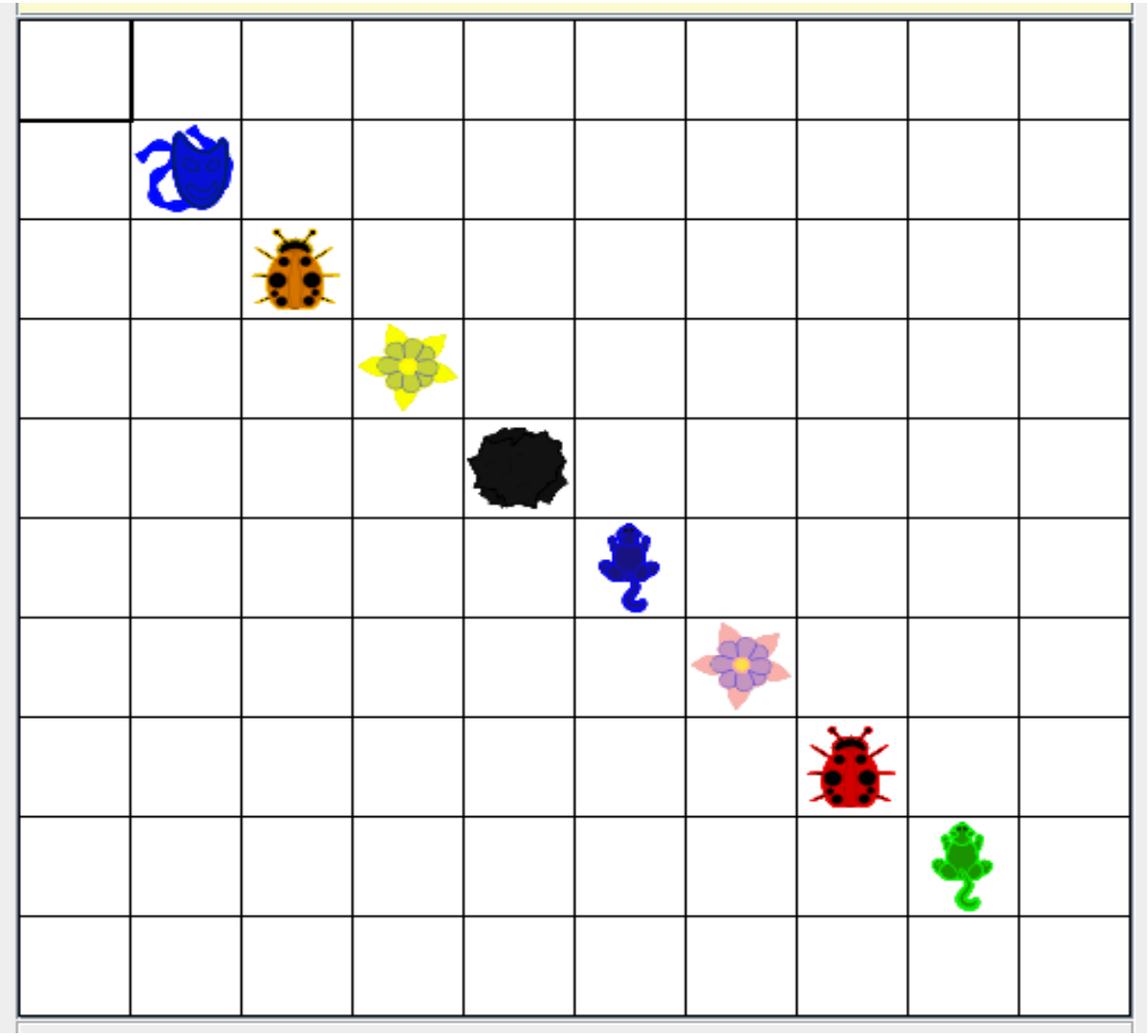
What is GridWorld?



Row = 2

Column = 1

What is GridWorld?



Grid is an interface that details the behaviors expected of a Grid.

Grid was designed as an interface because many different structures could be used to store the grid values.

An interface works perfectly due to the large number of unknowns.

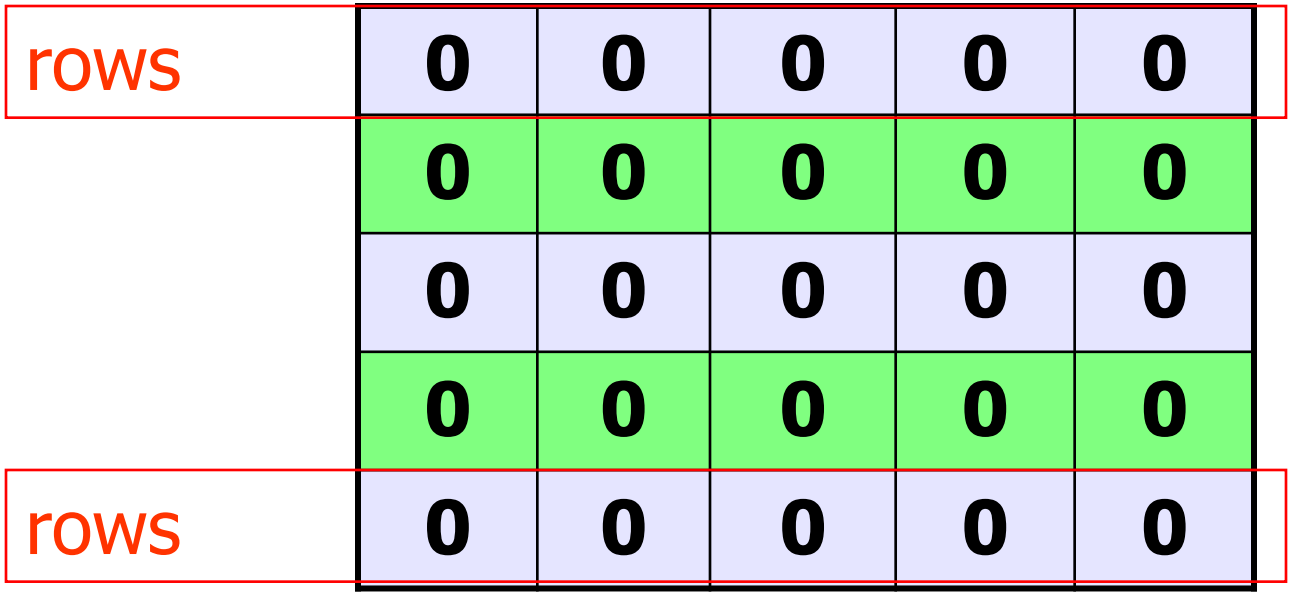
Grid

abstract methods

Name	Use
get(loc)	returns the ref at location loc
getEmptyAdjacentLocations(loc)	gets the valid empty locs in 8 dirs
getNeighbors(loc)	returns the objs around this in 8 dirs
getNumCols()	gets the # of cols for this grid
getNumRows()	gets the # of rows for this grid
getOccupiedAdjacentLocations(loc)	gets the valid locs in 8 dirs that contain objs
getOccupiedLocations()	gets locs that contain live objs
getValidAdjacentLocations(loc)	gets the valid locs in 8 dirs
isValid(loc)	checks to see if loc is valid
put(loc, obj)	put the obj in grid at location loc
remove(loc)	take the obj at location loc out of the grid

```
import info.gridworld.grid.Grid;
```

Grid

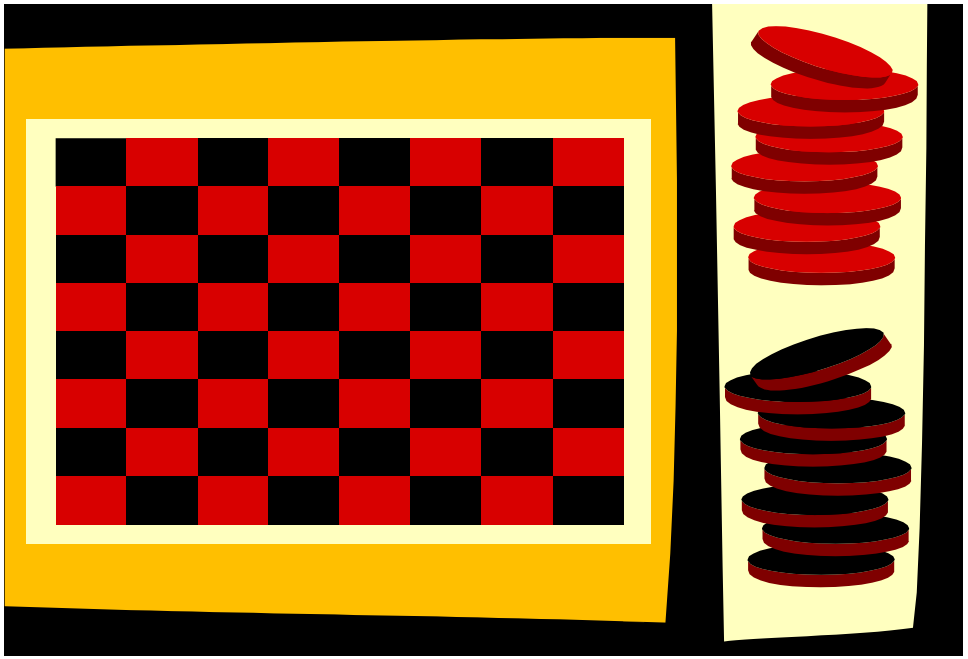


A grid is a structure that has rows and columns.

Grid

A grid is a structure that has rows and columns.

Grid



A grid is a structure that has rows and columns.

Critter

extends Actor

frequently used methods

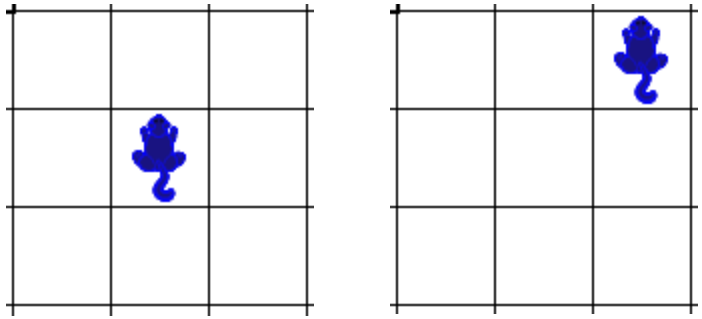
Name	Use
getColor()	gets the critter's color
getDirection()	gets the critter's direction
getLocation()	gets the critter's location
setColor(col)	sets the critter's color to col
setDirection(dir)	sets the critter's direction to dir

```
import info.gridworld.actor.Critter;
```

Critter

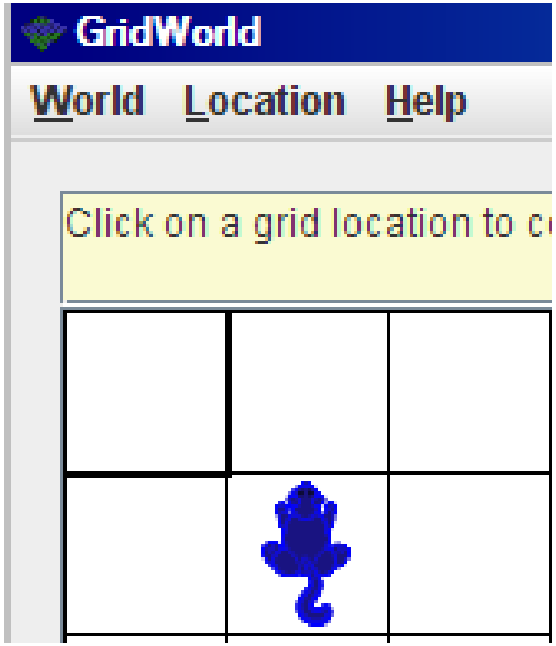
Critter differs from actor in that a critter moves around the grid and eats specific types of other actors.

Critter randomly picks one of its valid adjacent empty locations and moves to that location.



Critter

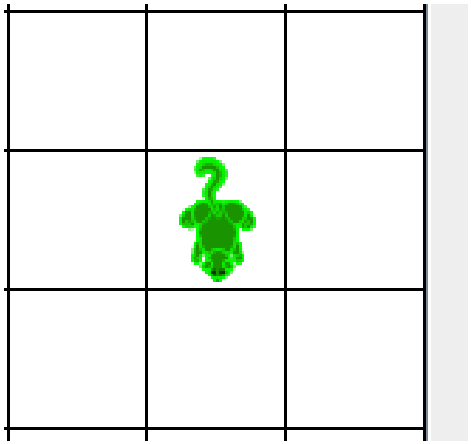
```
ActorWorld world = new ActorWorld();
Critter thang = new Critter();
world.add(new Location(1,1), thang);
world.show();
```



open critterone.java

Critter

```
ActorWorld world = new ActorWorld();
Critter thang = new Critter();
thang.setColor(Color.GREEN);
thang.setDirection(180);
Location loc = new Location(2,2);
world.add(loc, thang);
world.show();
```



open crittertwo.java

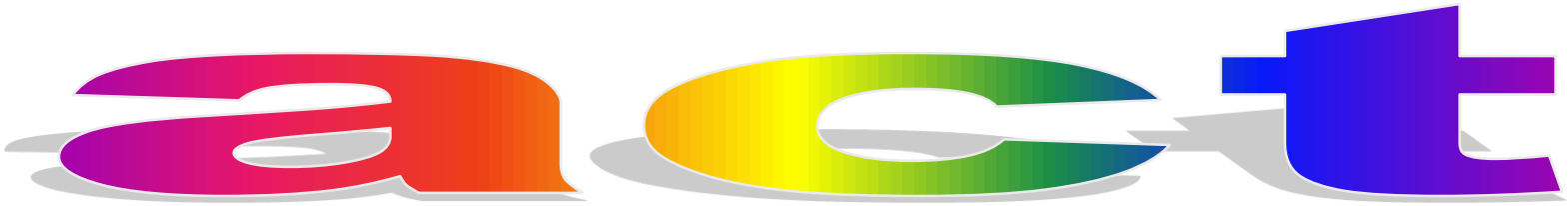
Critter

extends Actor

frequently used methods – Critter specific

Name	Use
act()	calls the methods listed below
getActors()	gets all actors around this location
processActors(actors)	do something to actors sent in
getMoveLocations()	gets list of possible move locs
selectMoveLocation(locs)	picks loc to move to
makeMove(loc)	moves this critter to loc

```
import info.gridworld.actor.Critter;
```



if no grid present – stop

call getActors to get list of actors to proces
processActors received from getActors

call getMoveLocations to get a list of locations
to which the critter might move
call selectMoveLocation to select new location

move to the new loc

getActors

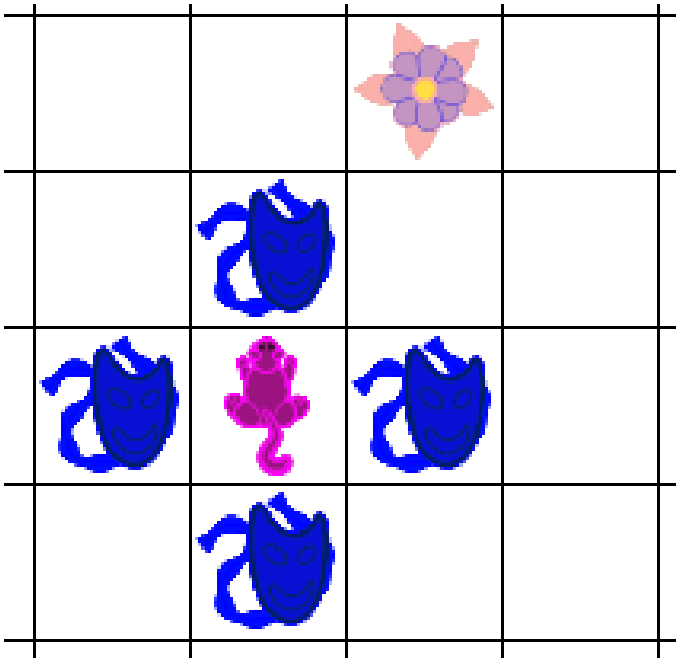
The getActors method returns an ArrayList containing all of the actors around this critter using the 4 cardinal(N,S,E,W) and 4 intercardinal directions(NE, NW, SE, SW).

In order to change which actors are returned by getActors, override the method and provide a different method of selecting actors.

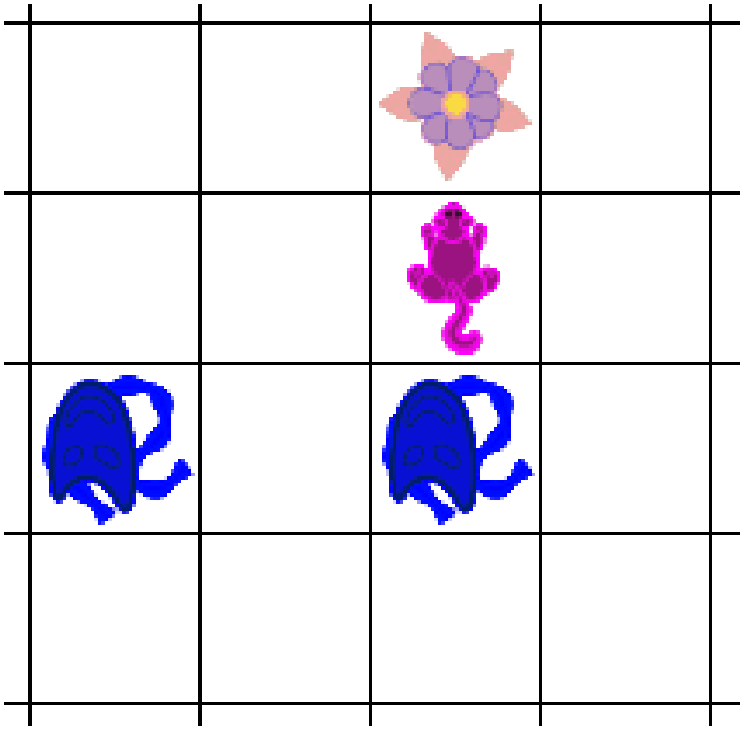
getActors must not modify any actors.

Extending Critter

Extending Critter



Use the GW quick reference!



What has to change if you want a critter to only get actors in front and behind?

Extending Critter

```
public class GetInFrontBehindCritter extends Critter
{
  //constructor

  public ArrayList<Actor> getActors()
  {




  }
}
```

What code is needed?

open getinfrontbehindcritter.java getinfrontbehindcritterrunner.java

processActors

The processActors method will do something to some or all of the actors around this critter.

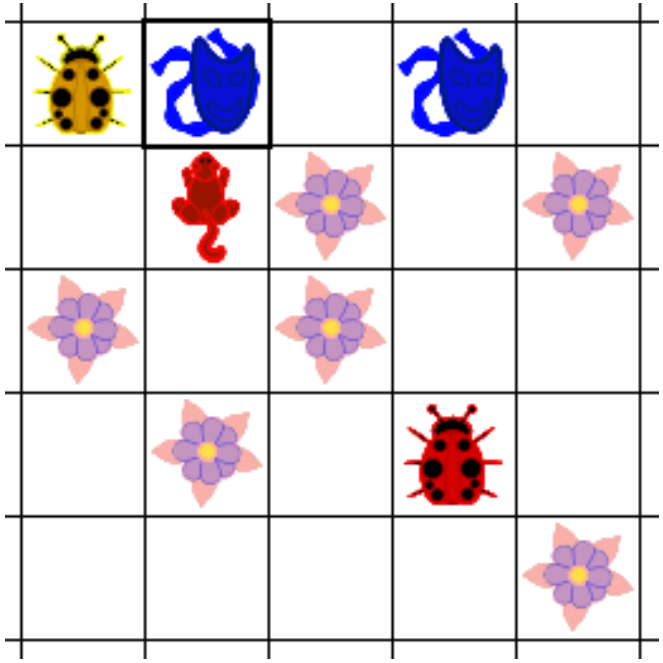
The processActors receives a list of all actors around this actor based on this actor's getActors method.

The critter act method calls getActors and passes the returned ArrayList to processActors.

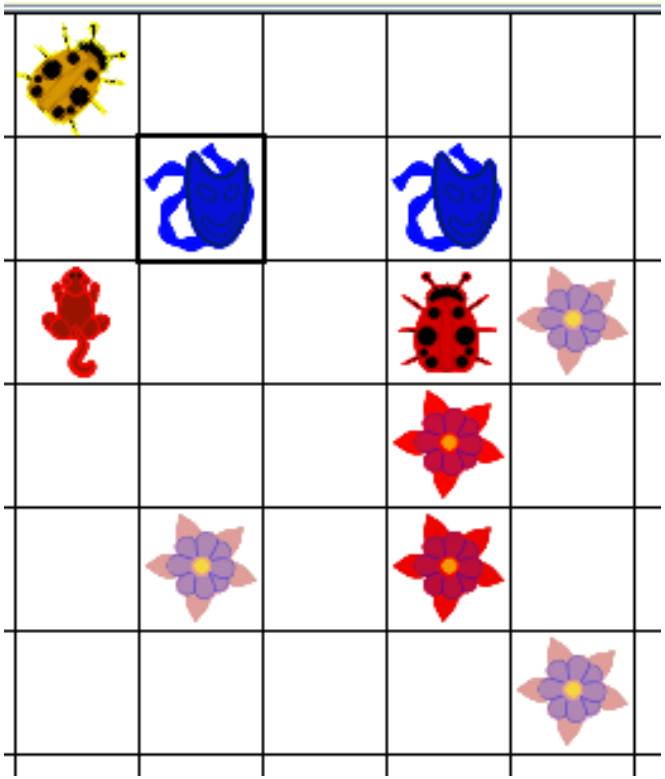
processActors must only change the actors received in the ArrayList parameter.

Extending Critter

Extending Critter



Use the GW quick reference!



What has to change if you want a critter to only eat flowers?

Extending Critter

```
public class FlowerEatingCritter extends Critter
{
  //constructor

  public void processActors(ArrayList<Actor> actors)
  {




  }
}
```

What code is needed?

open flowereatingcritter.java flowereatingcritterrunner.java

getMoveLocations

The getMoveLocations method returns a list of all empty adjacent locations to which this critter could move.

In order to change which locations are returned by getMoveLocations, override the method and provide a different method of selecting move locations.

getMoveLocations must not modify any actors.

selectMoveLocation

The selectMoveLocation method selects a possible move location from the list of locations returned by getMoveLocations.

The selectMoveLocation receives a list of all actors around this actor based on this actor's getMoveLocations method.

The critter act method calls getMoveLocations and passes the returned ArrayList to selectMoveLocation.

selectMoveLocation must not modify any actors.

makeMove

The makeMove method receives a location parameter.

If the parameter is null, the critter is removed from the grid.

If the parameter is not null, the critter moves to the new location. If an actor was in the location the critter is moving to, the actor is removed.

makeMove must only modify the actors at this critter's new and old locations.

open crabcritter.java chameleoncritter.java

Start work on Critter Labs and Exercises