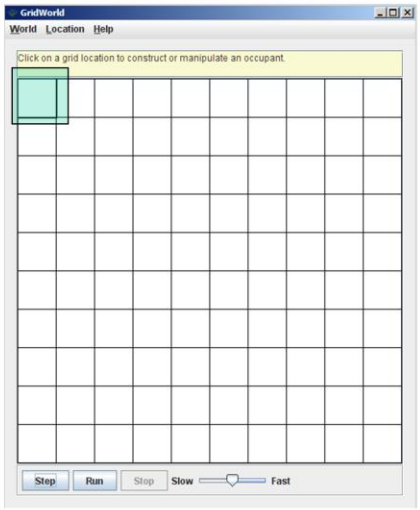

GridWorld

What is GridWorld?



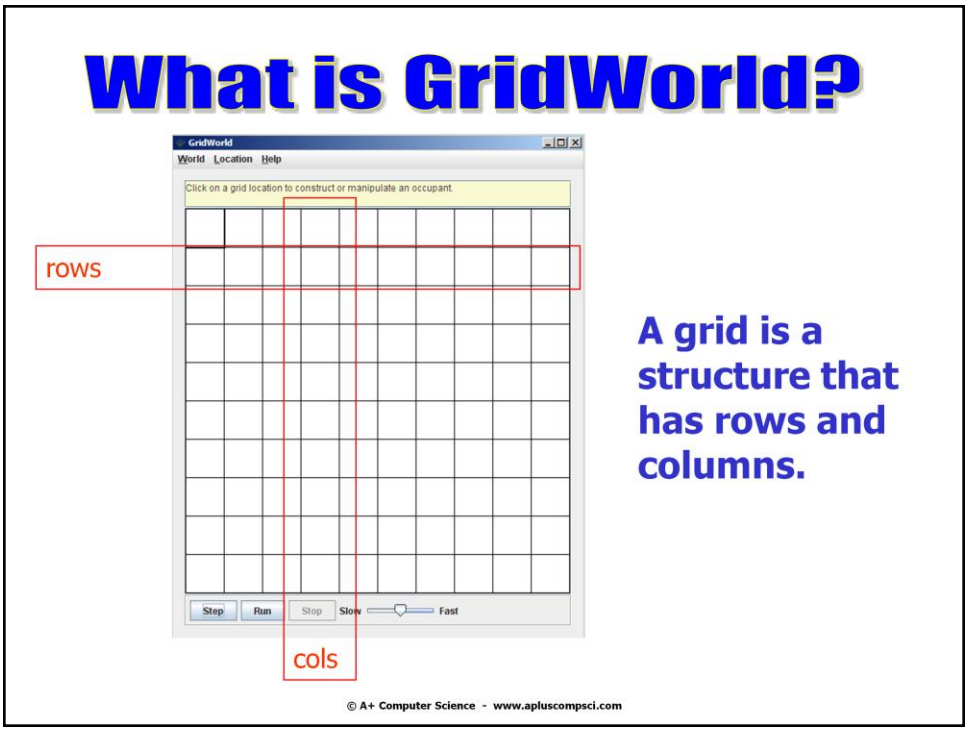
Row = 0

Column = 0

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

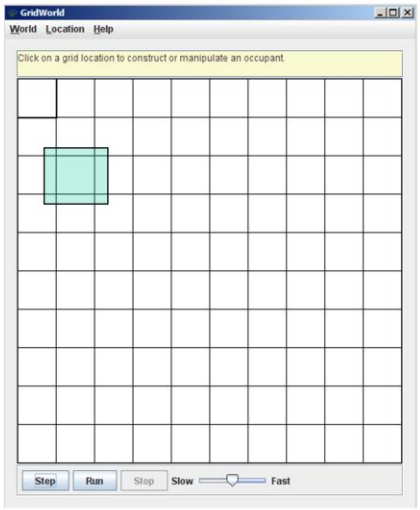


A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

What is GridWorld?



Row = 2

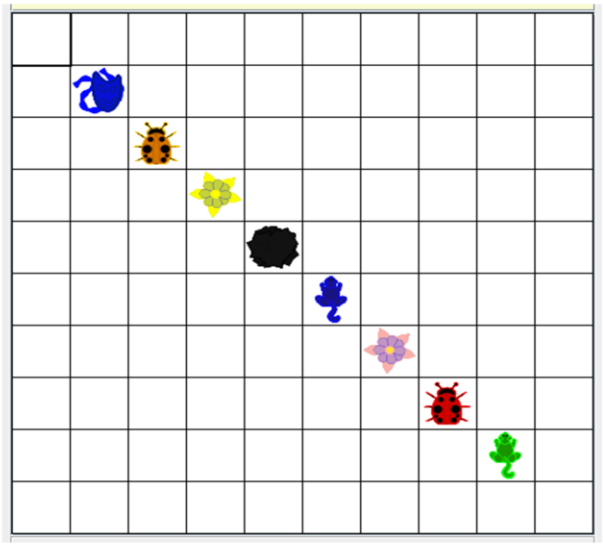
Column = 1

A grid is a structure that has rows and columns.

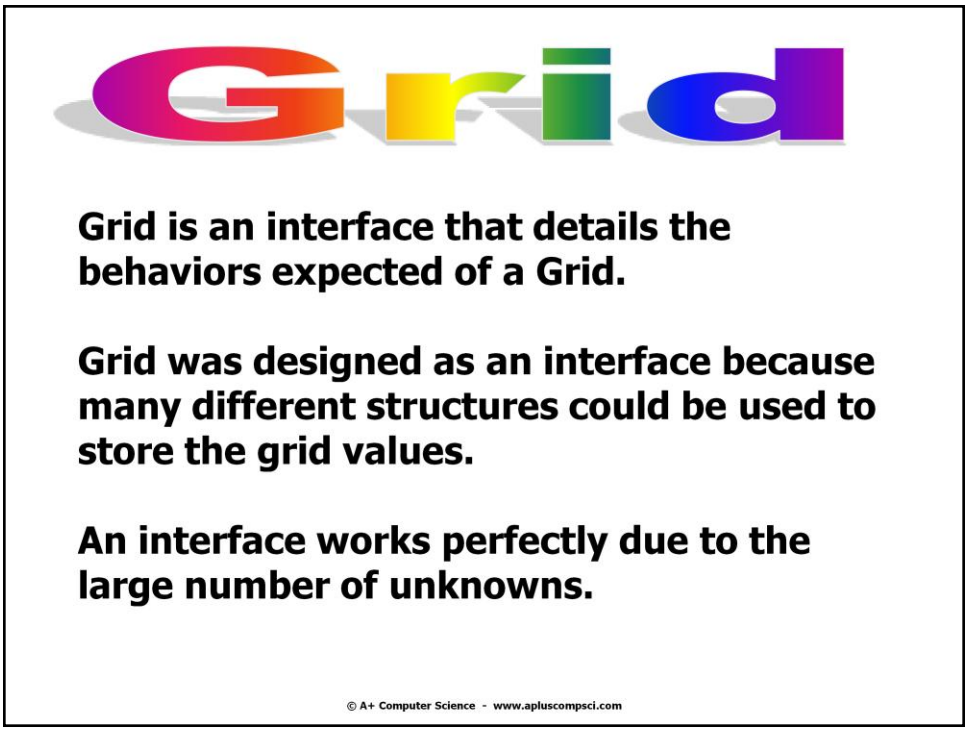
A spreadsheet is a grid.

A checker board is a grid.

What is GridWorld?




Grid
© A+ Computer Science - www.apluscompsci.com



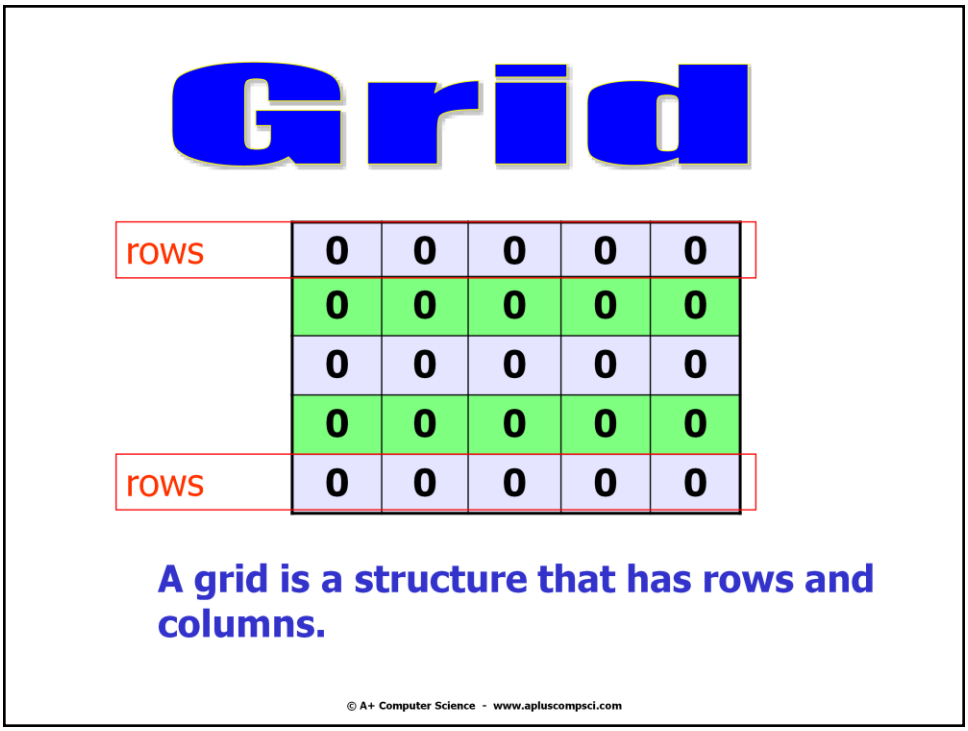
Grid is a row / column structure that stores Objects.

The location of each Object is determined by the Location provided when putting the Object in the grid.

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

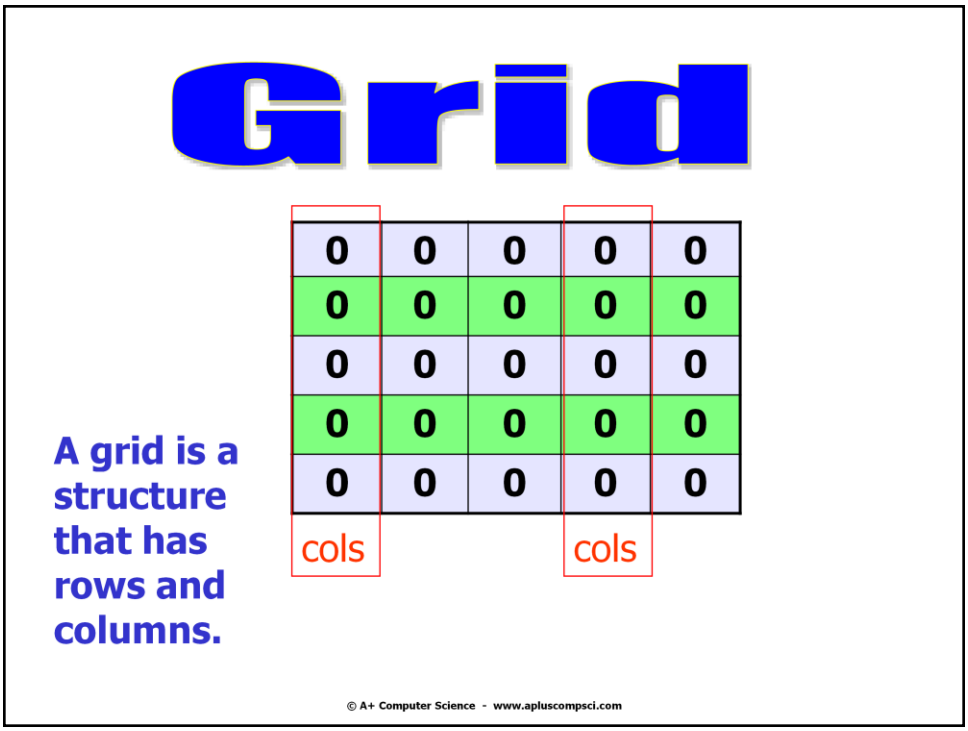
```
import info.gridworld.grid.Grid;
```



A grid is a structure that has rows and columns.

A spreadsheet is a grid.

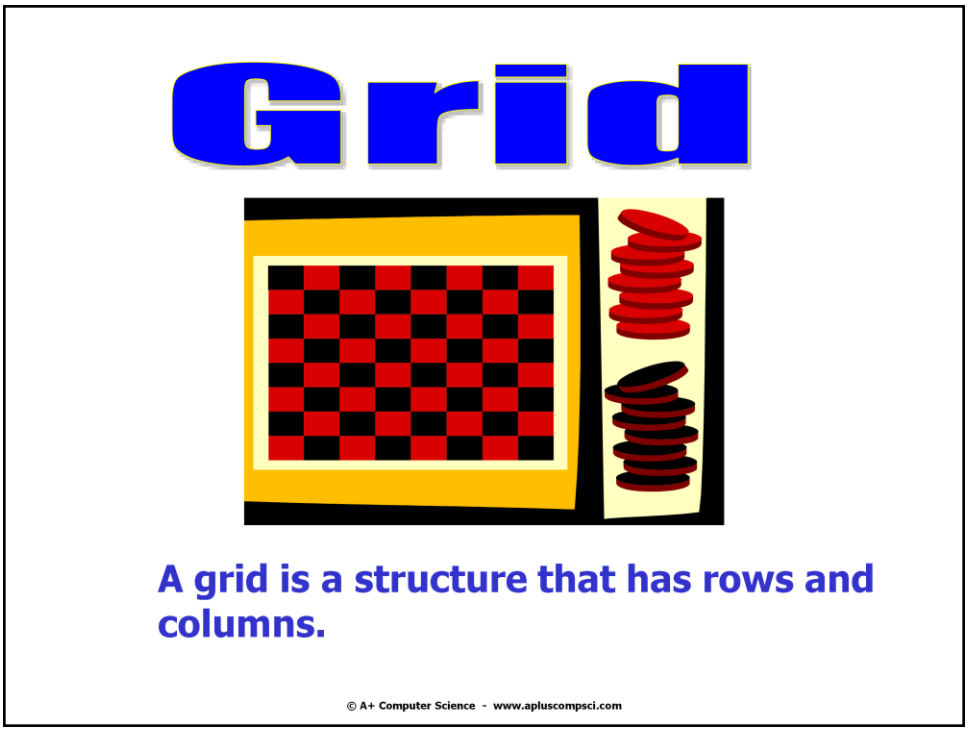
A checker board is a grid.



A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.



A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.


Critter
© A+ Computer Science - www.apluscompsci.com

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;
```


The methods listed below were inherited from actor.

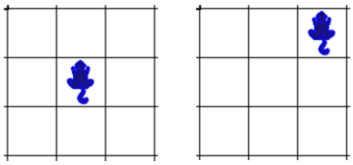
The act method has been overridden as the behavior of a bug is quite different from an actor.

The other methods listed above that were inherited have not been changed.

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.

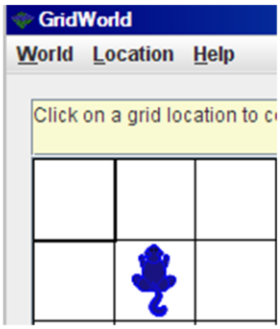


Critter is a suped up actor that moves around the grid and eats other actors.

Critter selects any of its neighbors that are empty and valid. Critter moves to the new location.

Critter

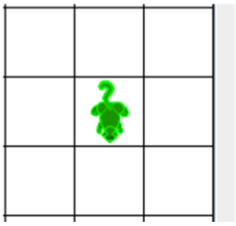
```
ActorWorld world = new ActorWorld();
Critter thang = new Critter();
world.add(new Location(1,1), thang);
world.show();
```



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;
```

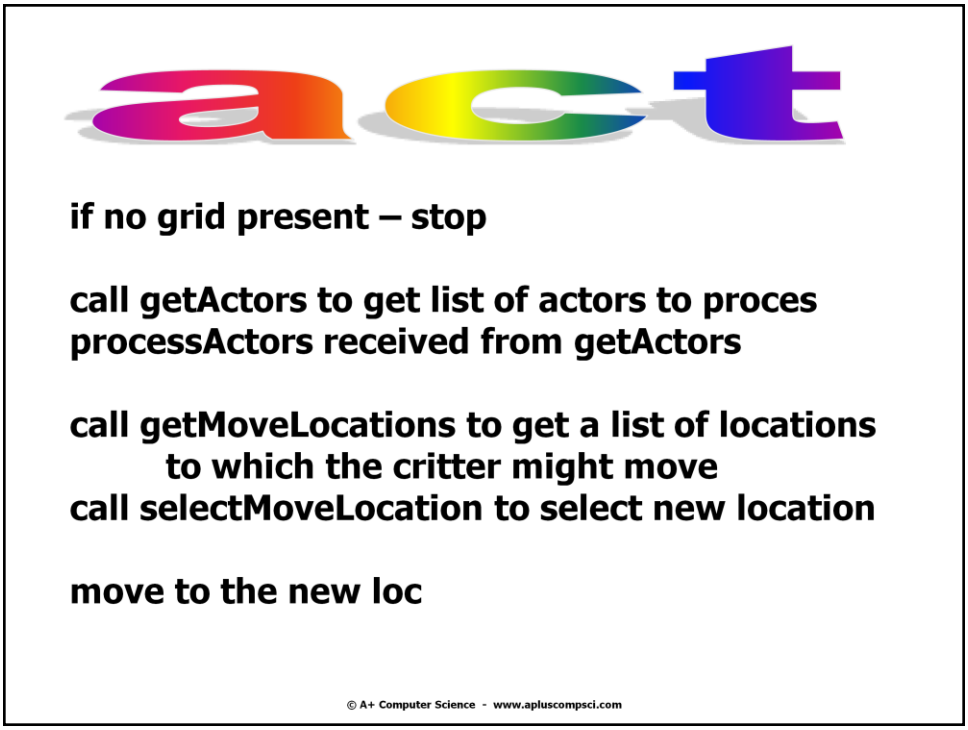

Critter's act method first calls the getActors method to get a list of actors around this critter.

Act then calls processActors and sends it the ArrayList of actors built by getActors. processActors typically does something to some or all of the actors in the list.

Act calls getMoveLocations next. getMoveLocations builds and returns an ArrayList of locations to which this critter could move.

selectMoveLocation is called and sent the ArrayList built by getMoveLocations. selectMoveLocation randomly picks one of the locations and returns it.

makeMove is called last by act and makes the critter move if possible.



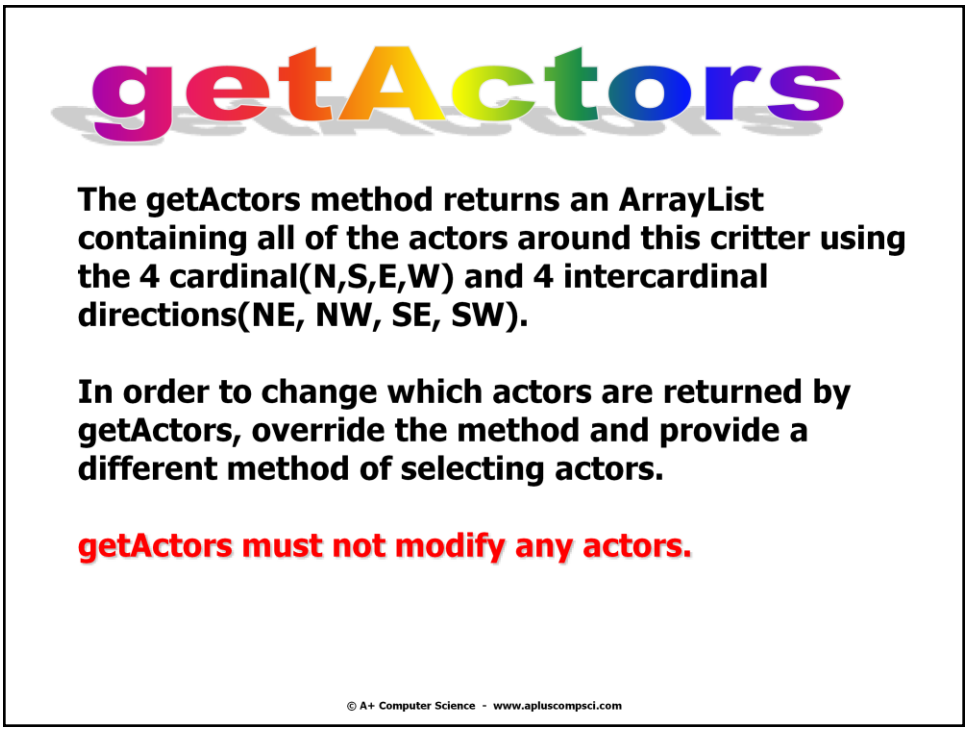
Critter's act method first calls the getActors method to get a list of actors around this critter.

Act then calls processActors and sends it the ArrayList of actors built by getActors. processActors typically does something to some or all of the actors in the list.

Act calls getMoveLocations next. getMoveLocations builds and returns an ArrayList of locations to which this critter could move.

selectMoveLocation is called and sent the ArrayList built by getMoveLocations. selectMoveLocation randomly picks one of the locations and returns it.

makeMove is called last by act and makes the critter move if possible.

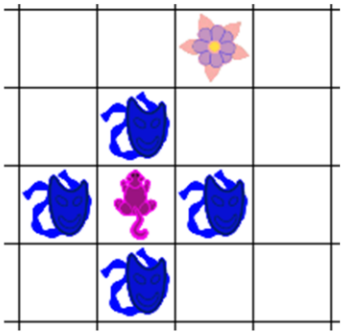


Critter's act method first calls the getActors method to get a list of actors around this critter.

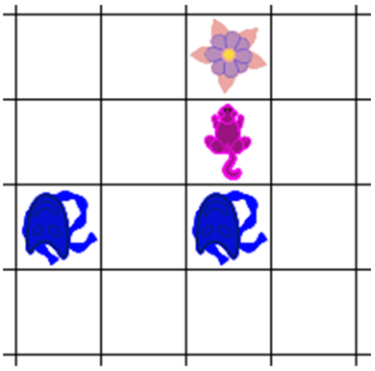
Act then calls processActors and sends it the ArrayList of actors built by getActors. processActors typically does something to some or all of the actors.


Extending
Critter
© A+ Computer Science - www.apluscompsci.com

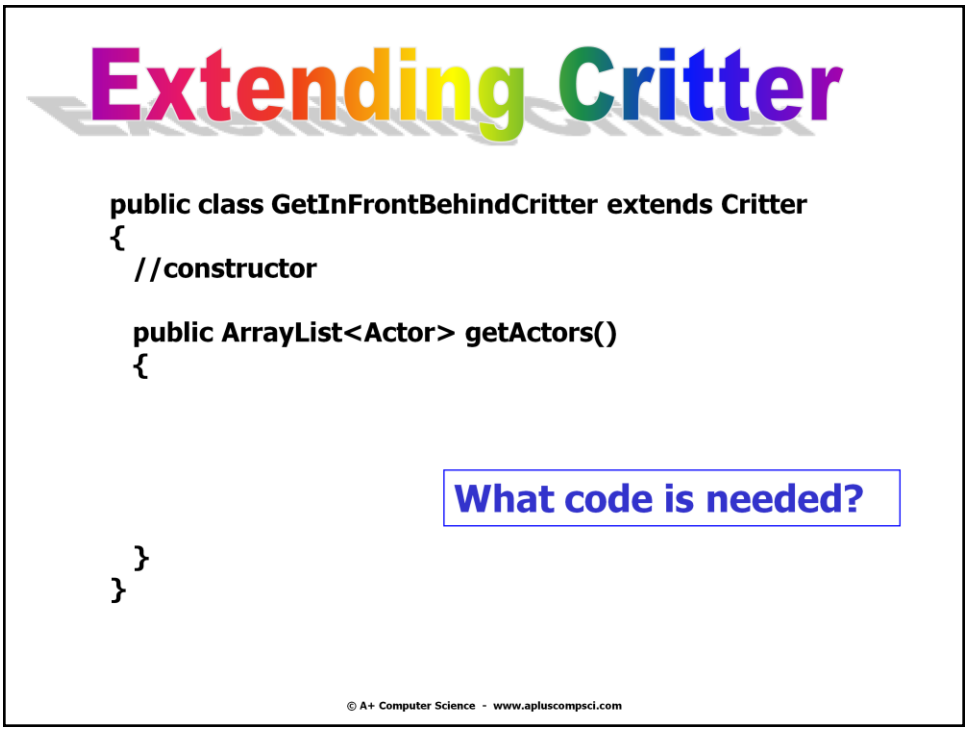
Extending Critter



Use the GW quick reference!

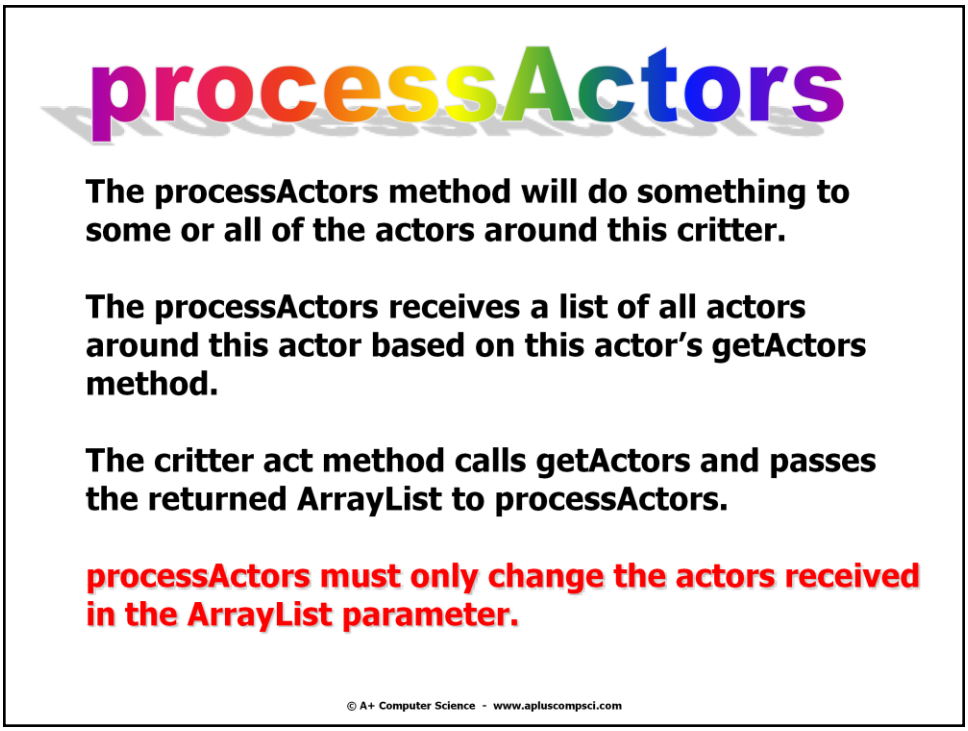


What has to change if you want a critter to only get actors in front and behind?



The original critter will select all adjacent actors.

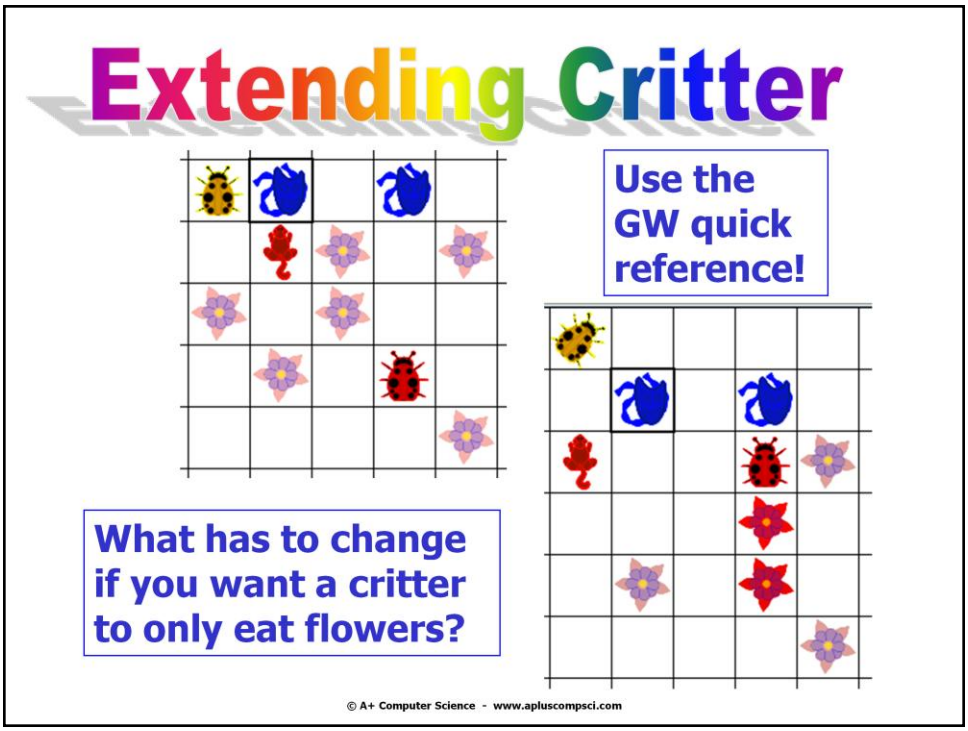
open
getinfrontbehindcritter.java
getinfrontbehindcritterrunner.java



Critter's act method first calls the getActors method to get a list of actors around this critter.

Act then calls processActors and sends it the ArrayList of actors built by getActors. processActors typically does something to some or all of the actors.


Extending
Critter
© A+ Computer Science - www.apluscompsci.com



The original critter eats everything except rocks and other critters.

Which method is used to manipulate the other actors around this critter? processActors

How much of processActors has to change so that the critter eats only rocks?

Extending Critter

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

  public void processActors(ArrayList<Actor> actors)
  {



  }
}
```

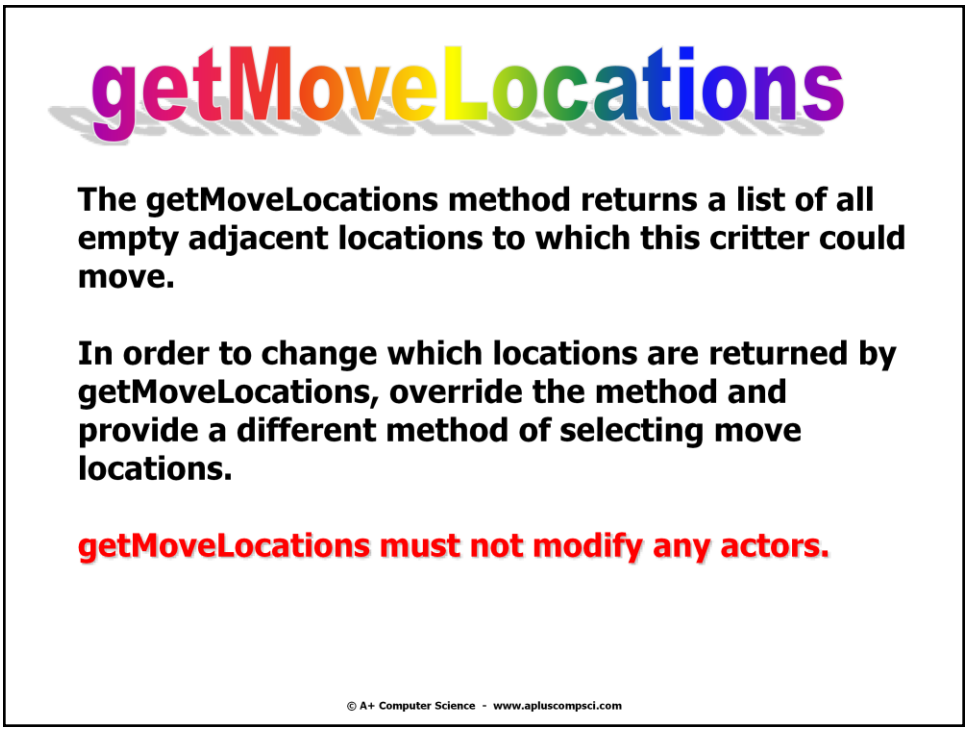
What code is needed?

The original critter eats everything except rocks and other critters.

Which method is used to manipulate the other actors around this critter? processActors

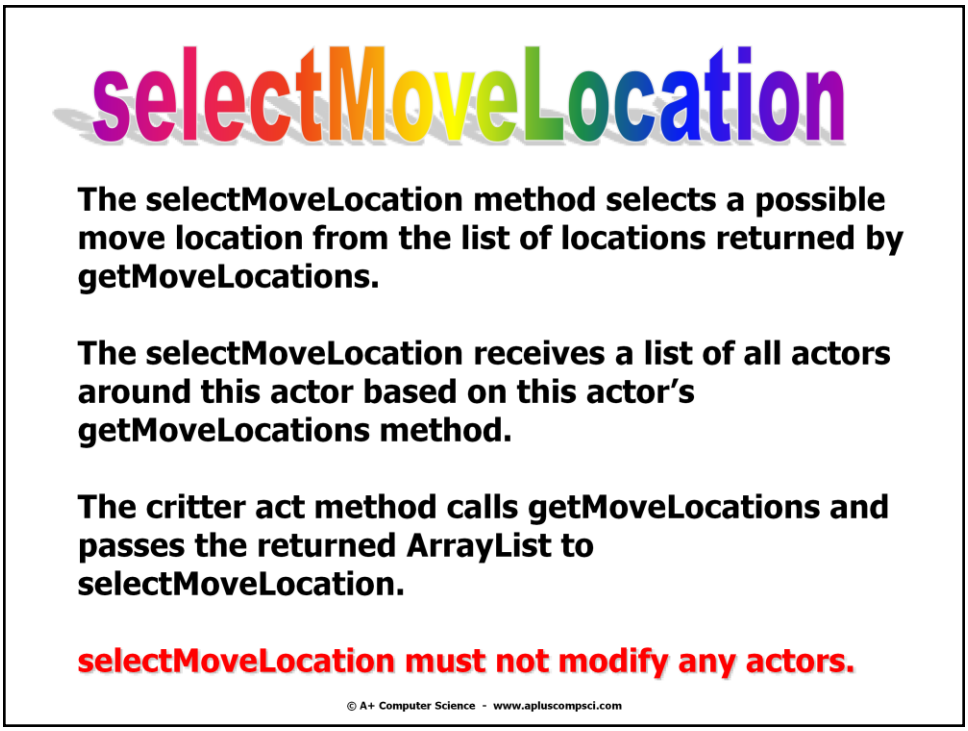
How much of processActors has to change so that the critter eats only rocks?

open
flowereatingcritter.java
flowereatingcritterrunner.java



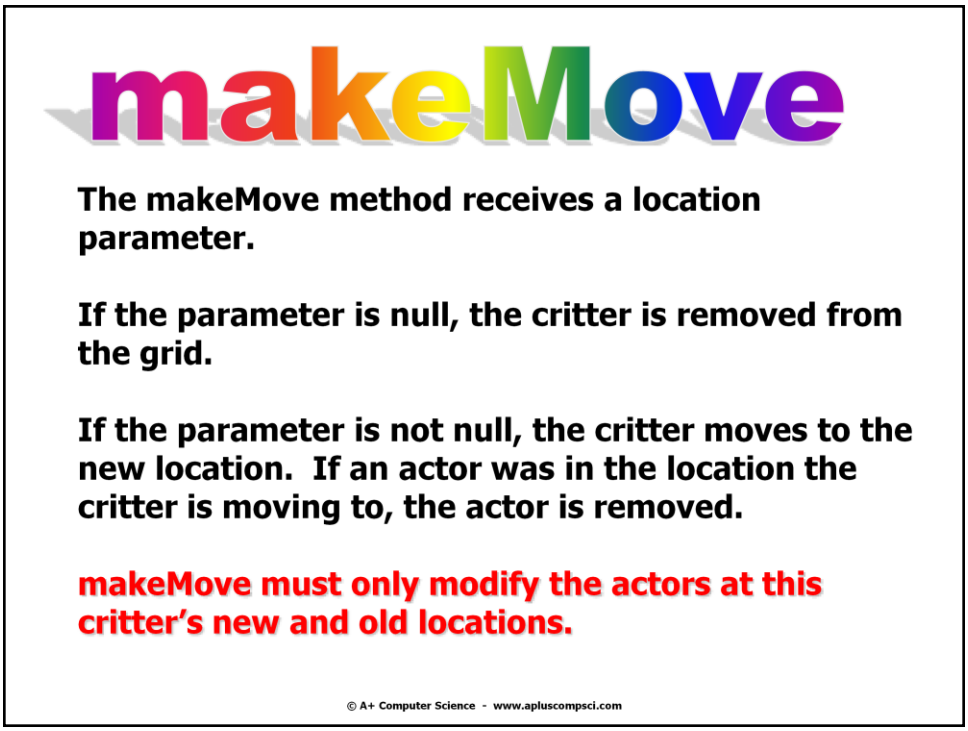
Act calls getMoveLocations next. getMoveLocations builds and returns an ArrayList of locations to which this critter could move.

selectMoveLocation is called and sent the ArrayList built by getMoveLocations. selectMoveLocation randomly picks one of the locations and returns it.



Act calls getMoveLocations next. getMoveLocations builds and returns an ArrayList of locations to which this critter could move.

selectMoveLocation is called and sent the ArrayList built by getMoveLocations. selectMoveLocation randomly picks one of the locations and returns it.



- makeMove is called last by act and makes the critter move if possible.

open
crabcritter.java
chameleoncritter.java

Start work on Critter Labs and Exercises