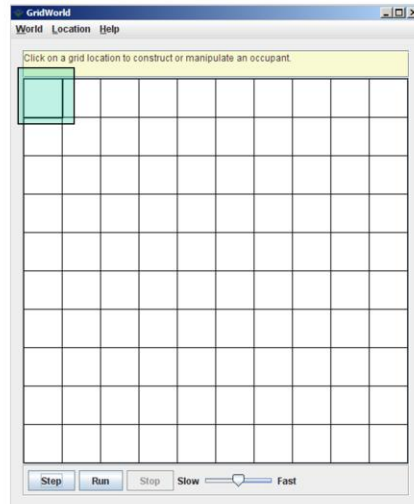




# What is GridWorld?



**Row = 0**

**Column = 0**

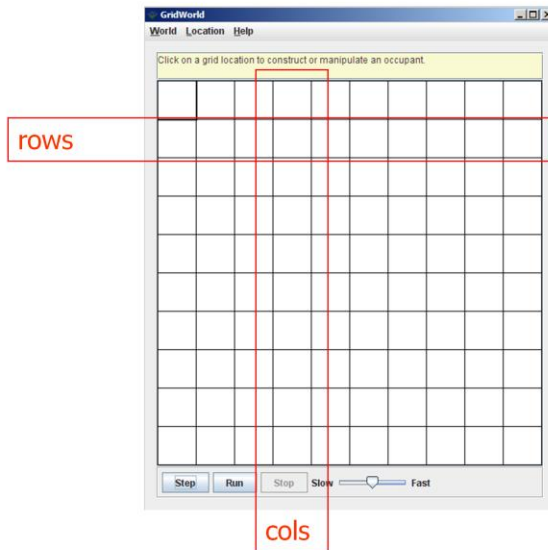
© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

# What is GridWorld?



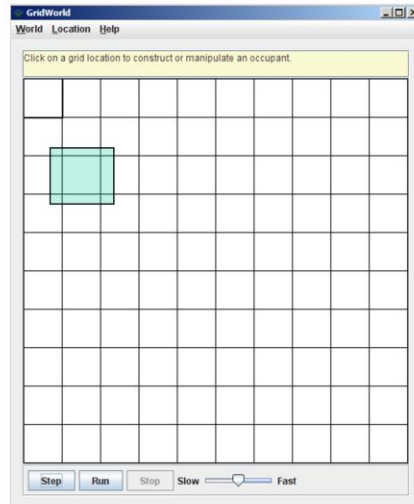
**A grid is a structure that has rows and columns.**

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**

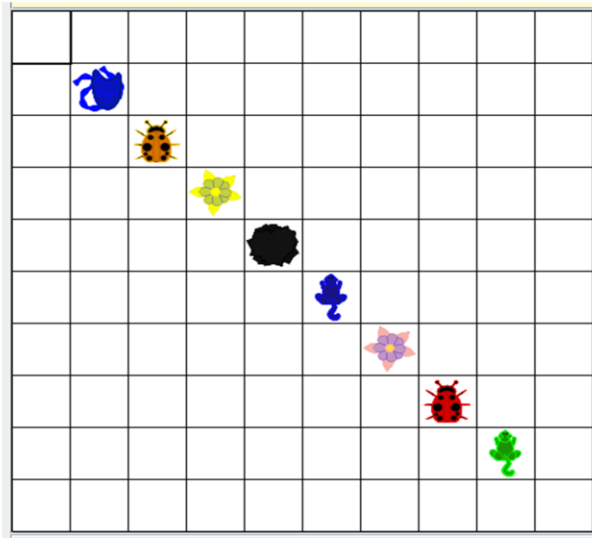
© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

# What is GridWorld?



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)



**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.**

© A+ Computer Science - [www.apluscompsci.com](http://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
<code>get(loc)</code>	returns the ref at location loc
<code>getEmptyAdjacentLocations(loc)</code>	gets the valid empty locs in 8 dirs
<code>getNeighbors(loc)</code>	returns the objs around this in 8 dirs
<code>getNumCols()</code>	gets the # of cols for this grid
<code>getNumRows()</code>	gets the # of rows for this grid
<code>getOccupiedAdjacentLocations(loc)</code>	gets the valid locs in 8 dirs that contain objs
<code>getOccupiedLocations()</code>	gets locs that contain live objs
<code>getValidAdjacentLocations(loc)</code>	gets the valid locs in 8 dirs
<code>isValid(loc)</code>	checks to see if loc is valid
<code>put(loc, obj)</code>	put the obj in grid at location loc
<code>remove(loc)</code>	take the obj at location loc out of the grid

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

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)



# Grid

rows	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
	0	0	0	0	0
rows	0	0	0	0	0

**A grid is a structure that has rows and columns.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

# Grid

**A grid is a structure that has rows and columns.**

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
cols		cols		

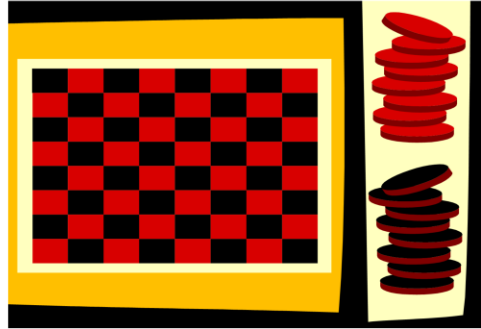
© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.

# Grid



**A grid is a structure that has rows and columns.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

A grid is a structure that has rows and columns.

A spreadsheet is a grid.

A checker board is a grid.



© A+ Computer Science - [www.apluscompsci.com](http://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;`

© A+ Computer Science - www.apluscompsci.com

The methods listed below were inherited from actor.

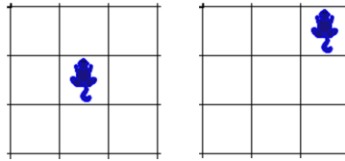
The act method has been overridden as the behavior of a bug is quite different from an actor.

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.**



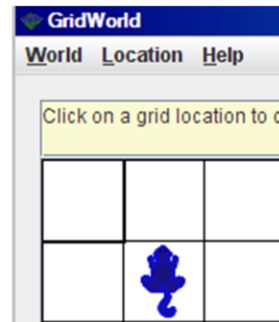
© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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();
```



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

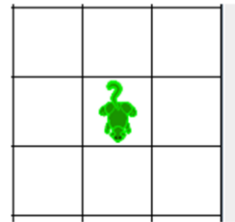
**open  
critterone.java**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)



# 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();
```



© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

**open**  
**crittertwo.java**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

<p style="text-align: center;"><b>Critter</b></p> <p style="text-align: center;">extends Actor</p> <p style="text-align: center;"><b>frequently used methods – Critter specific</b></p>	
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;**

© A+ Computer Science - www.apluscompsci.com

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.



**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**

© A+ Computer Science - www.apluscompsci.com

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.

# 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.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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

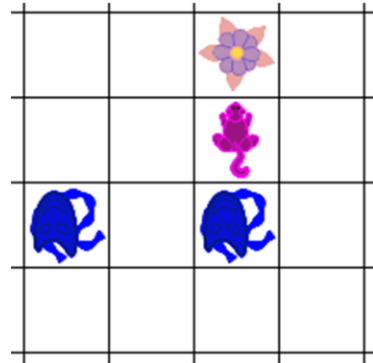
Act then calls processActors and sends it the ArrayList of actors built by getActors. processActors typically does something to some or all of the actors.



# Extending Critter



Use the  
GW quick  
reference!



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

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

# Extending Critter

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

    public ArrayList<Actor> getActors()
    {

    }

}
```

What code is needed?

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

The original critter will select all adjacent actors.



```
open  
getinfrontbehindcritter.java  
getinfrontbehindcritterrunner.java
```

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

# 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.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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 Criticter

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

# Extending Critter

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

**Use the GW quick reference!**

© A+ Computer Science - [www.apluscompsci.com](http://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?

© A+ Computer Science - [www.apluscompsci.com](http://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?

**open**  
**flowereatingcritter.java**  
**flowereatingcritterrunner.java**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

# 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.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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.

# 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.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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

**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.**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

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

**open**  
**crabcritter.java**  
**chameleonecritter.java**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)

# **Start work on Critter Labs and Exercises**

© A+ Computer Science - [www.apluscompsci.com](http://www.apluscompsci.com)