| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/info/gridworld/actor/Bug.html)   [**NEXT CLASS**](http://docs.google.com/info/gridworld/actor/Flower.html) | [**FRAMES**](http://docs.google.com/index.html?info/gridworld/actor/Critter.html)    [**NO FRAMES**](http://docs.google.com/Critter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **info.gridworld.actor**

Class Critter

[java.lang.Object](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true)  
 [info.gridworld.actor.Actor](http://docs.google.com/info/gridworld/actor/Actor.html)  
 **info.gridworld.actor.Critter**

public class **Critter**extends [Actor](http://docs.google.com/info/gridworld/actor/Actor.html)

A Critter is an actor that moves through its world, processing other actors in some way and then picking a new location.

The implementation of this class is testable on the AP CS A and AB exams.

| **Constructor Summary** | |
| --- | --- |
| [**Critter**](http://docs.google.com/info/gridworld/actor/Critter.html#Critter())() |

| **Method Summary** | |
| --- | --- |
| void | [**act**](http://docs.google.com/info/gridworld/actor/Critter.html#act())()            A critter acts by getting a list of its neighbors, processing them, getting locations to move to, selecting one of them, and moving to the selected location. |
| [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Actor](http://docs.google.com/info/gridworld/actor/Actor.html)> | [**getActors**](http://docs.google.com/info/gridworld/actor/Critter.html#getActors())()            Gets the actors for processing. |
| [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> | [**getMoveLocations**](http://docs.google.com/info/gridworld/actor/Critter.html#getMoveLocations())()            Gets the possible locations for the next move. |
| void | [**makeMove**](http://docs.google.com/info/gridworld/actor/Critter.html#makeMove(info.gridworld.grid.Location))([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)            Moves this critter to the given location. |
| void | [**processActors**](http://docs.google.com/info/gridworld/actor/Critter.html#processActors(java.util.ArrayList))([ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Actor](http://docs.google.com/info/gridworld/actor/Actor.html)> actors)            Processes the actors. |
| [Location](http://docs.google.com/info/gridworld/grid/Location.html) | [**selectMoveLocation**](http://docs.google.com/info/gridworld/actor/Critter.html#selectMoveLocation(java.util.ArrayList))([ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> locs)            Selects the location for the next move. |

| **Methods inherited from class info.gridworld.actor.**[**Actor**](http://docs.google.com/info/gridworld/actor/Actor.html) |
| --- |
| [getColor](http://docs.google.com/info/gridworld/actor/Actor.html#getColor()), [getDirection](http://docs.google.com/info/gridworld/actor/Actor.html#getDirection()), [getGrid](http://docs.google.com/info/gridworld/actor/Actor.html#getGrid()), [getLocation](http://docs.google.com/info/gridworld/actor/Actor.html#getLocation()), [moveTo](http://docs.google.com/info/gridworld/actor/Actor.html#moveTo(info.gridworld.grid.Location)), [putSelfInGrid](http://docs.google.com/info/gridworld/actor/Actor.html#putSelfInGrid(info.gridworld.grid.Grid,%20info.gridworld.grid.Location)), [removeSelfFromGrid](http://docs.google.com/info/gridworld/actor/Actor.html#removeSelfFromGrid()), [setColor](http://docs.google.com/info/gridworld/actor/Actor.html#setColor(java.awt.Color)), [setDirection](http://docs.google.com/info/gridworld/actor/Actor.html#setDirection(int)), [toString](http://docs.google.com/info/gridworld/actor/Actor.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true) |
| --- |
| [clone](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#clone()), [equals](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#equals(java.lang.Object)), [finalize](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#finalize()), [getClass](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#getClass()), [hashCode](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#hashCode()), [notify](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#notify()), [notifyAll](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#notifyAll()), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait()), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait(long)), [wait](http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Object.html?is-external=true#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### Critter

public **Critter**()

| **Method Detail** |
| --- |

### act

public void **act**()

A critter acts by getting a list of its neighbors, processing them, getting locations to move to, selecting one of them, and moving to the selected location.

**Overrides:**[act](http://docs.google.com/info/gridworld/actor/Actor.html#act()) in class [Actor](http://docs.google.com/info/gridworld/actor/Actor.html)

### getActors

public [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Actor](http://docs.google.com/info/gridworld/actor/Actor.html)> **getActors**()

Gets the actors for processing. The actors must be contained in the same grid as this critter. Implemented to return the actors that occupy neighboring grid locations. Override this method in subclasses to look elsewhere for actors to process.

**Returns:**a list of actors that are neighbors of this critter

### processActors

public void **processActors**([ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Actor](http://docs.google.com/info/gridworld/actor/Actor.html)> actors)

Processes the actors. Implemented to "eat" (i.e. remove) all actors that are not rocks or critters. Override this method in subclasses to process neighbors in a different way.

Precondition: All objects in actors are contained in the same grid as this critter.

**Parameters:**actors - the actors to be processed

### getMoveLocations

public [ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> **getMoveLocations**()

Gets the possible locations for the next move. Implemented to return the empty neighboring locations. Override this method in subclasses to look elsewhere for move locations.

Postcondition: The locations must be valid in the grid of this critter.

**Returns:**a list of possible locations for the next move

### selectMoveLocation

public [Location](http://docs.google.com/info/gridworld/grid/Location.html) **selectMoveLocation**([ArrayList](http://java.sun.com/j2se/1.5.0/docs/api/java/util/ArrayList.html?is-external=true)<[Location](http://docs.google.com/info/gridworld/grid/Location.html)> locs)

Selects the location for the next move. Implemented to randomly pick one of the possible locations, or to return the current location if locs has size 0. Override this method in subclasses that have another mechanism for selecting the next move location.

Precondition: All locations in locs are valid in the grid of this critter

**Parameters:**locs - the possible locations for the next move **Returns:**the location that was selected for the next move.

### makeMove

public void **makeMove**([Location](http://docs.google.com/info/gridworld/grid/Location.html) loc)

Moves this critter to the given location. Implemented to call moveTo. Override this method in subclasses that want to carry out other actions for moving (for example, turning or leaving traces).

Precondition: loc is valid in the grid of this critter

**Parameters:**loc - the location to move to (must be valid)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-all.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | | |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/info/gridworld/actor/Bug.html)   [**NEXT CLASS**](http://docs.google.com/info/gridworld/actor/Flower.html) | [**FRAMES**](http://docs.google.com/index.html?info/gridworld/actor/Critter.html)    [**NO FRAMES**](http://docs.google.com/Critter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |