GRIDWORLD CASE STUDY – PART IV

CRITTERS : COMPLEX ACTORS

I. GET A LIST OF ACTORS TO PROCESS

II. PROCESS THE ACTORS

III. GENERATE SET OF LOCATIONS TO WHICH IT MAY MOVE

IV. SELECTS ONE AND MOVES TO THAT LOCATION

AN EXAMPLE

I. A CRITTER COULD: GETS A LIST OF ALL NEIGHBORING ACTORS
- GET A LIST OF ALL NEIGHBORING ACTORS
- CHANGE THEIR COLOR
- MOVE TO A NEW EMPTY LOCATION.

THE act method IN THE Critter class (subclass of Actor) invokes the following five methods:

CRITTER METHOD	WHAT THE BASE METHOD DOES.
ArrayList<Actor> getActors( )	Gets a list of all neighboring actors.
void processActors(ArrayList<Actor> actors)	Eats (removes) actors that are not rocks or critters.
ArrayList<Location> getMoveLocations( )	Gets a list of all empty adjacent locations.
Location selectMoveLocation(ArrayList<Location> locs)	Selects an empty location at random.
void makeMove(Location loc)	Calls moveTo method to move into selected location.

YOUR SUBCLASSES OF Critter SHOULD override one or more of these methods.

PRE-WORK

How is the behavior of the ChameleonCritter / CrabCritter different from the base Critter?

What methods have been overridden? Why?

Do the exercises on page 35. Stick to this DEVELOPMENT RECIPE:

- Establish what the requirements will be: processing, movement.
- Develop test criteria
- Decide which methods will have to be modified
- Modify the methods, one at a time.
- Create test environment (put rocks, flowers, other critters on world)
- Perform the tests.
- Rework.