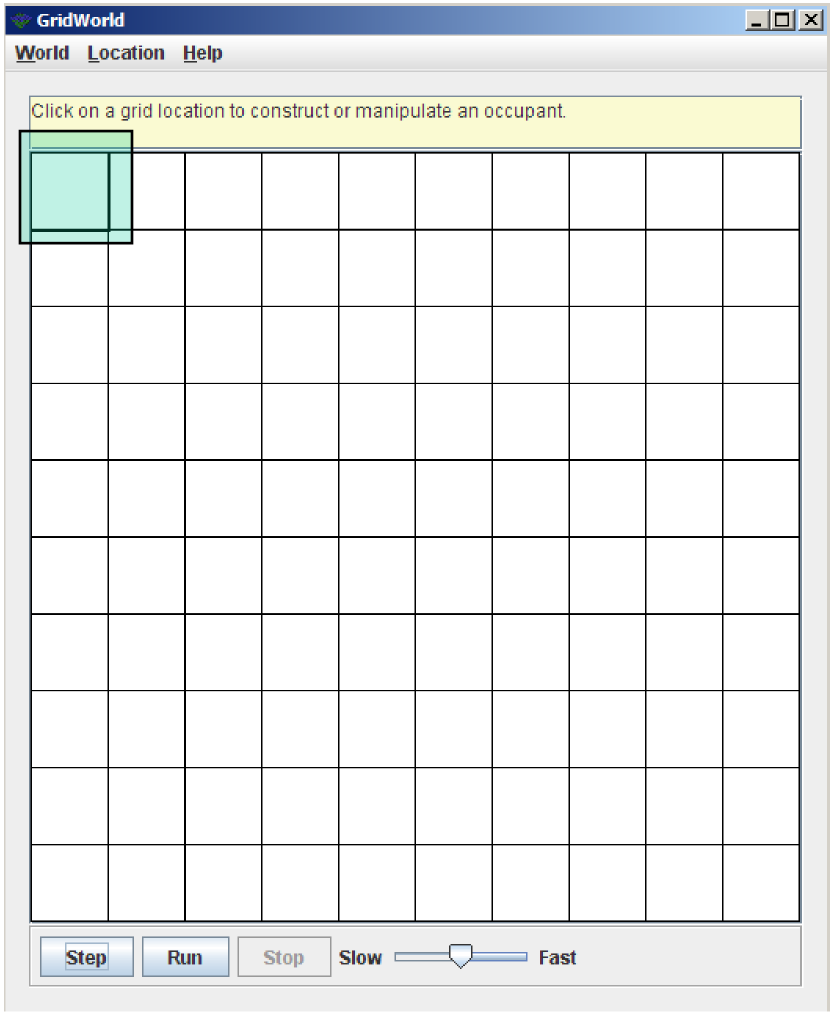
GridWorld

What is GridWorld?

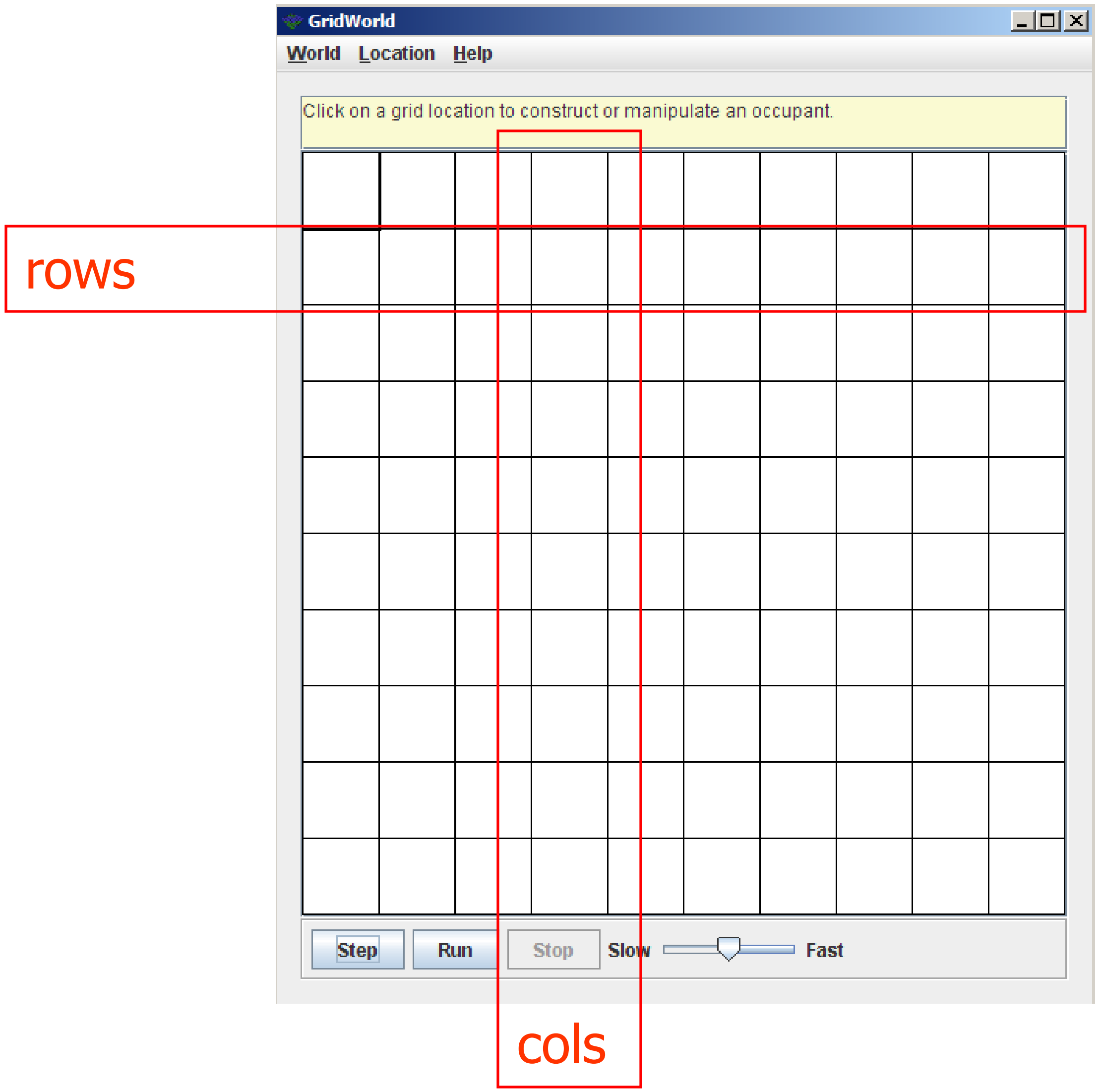


Click on a grid location to construct or manipulate an occupant.

Row = 0

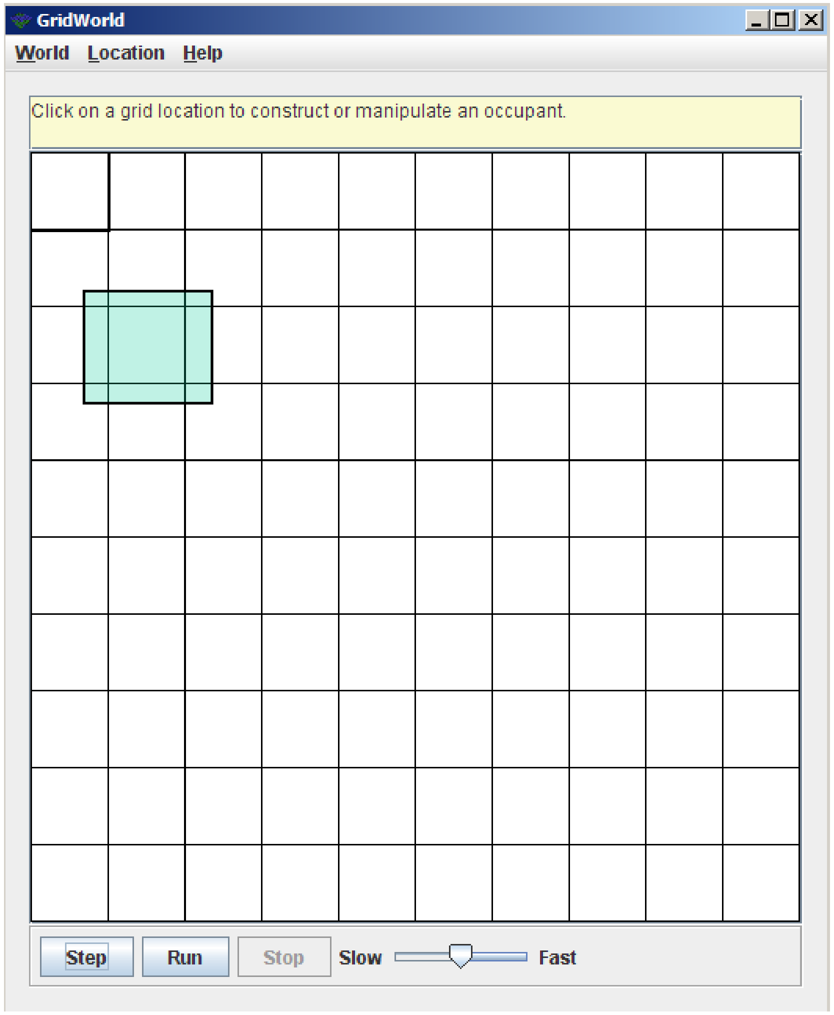
Column = 0

What is GridWorld?


GridWorld
World
Location
Help
Click on a grid location to construct or manipulate an occupant.
rows
cols
Step
Run
Stop
Slow
Fast

A grid is a structure that has rows and columns.

What is GridWorld?

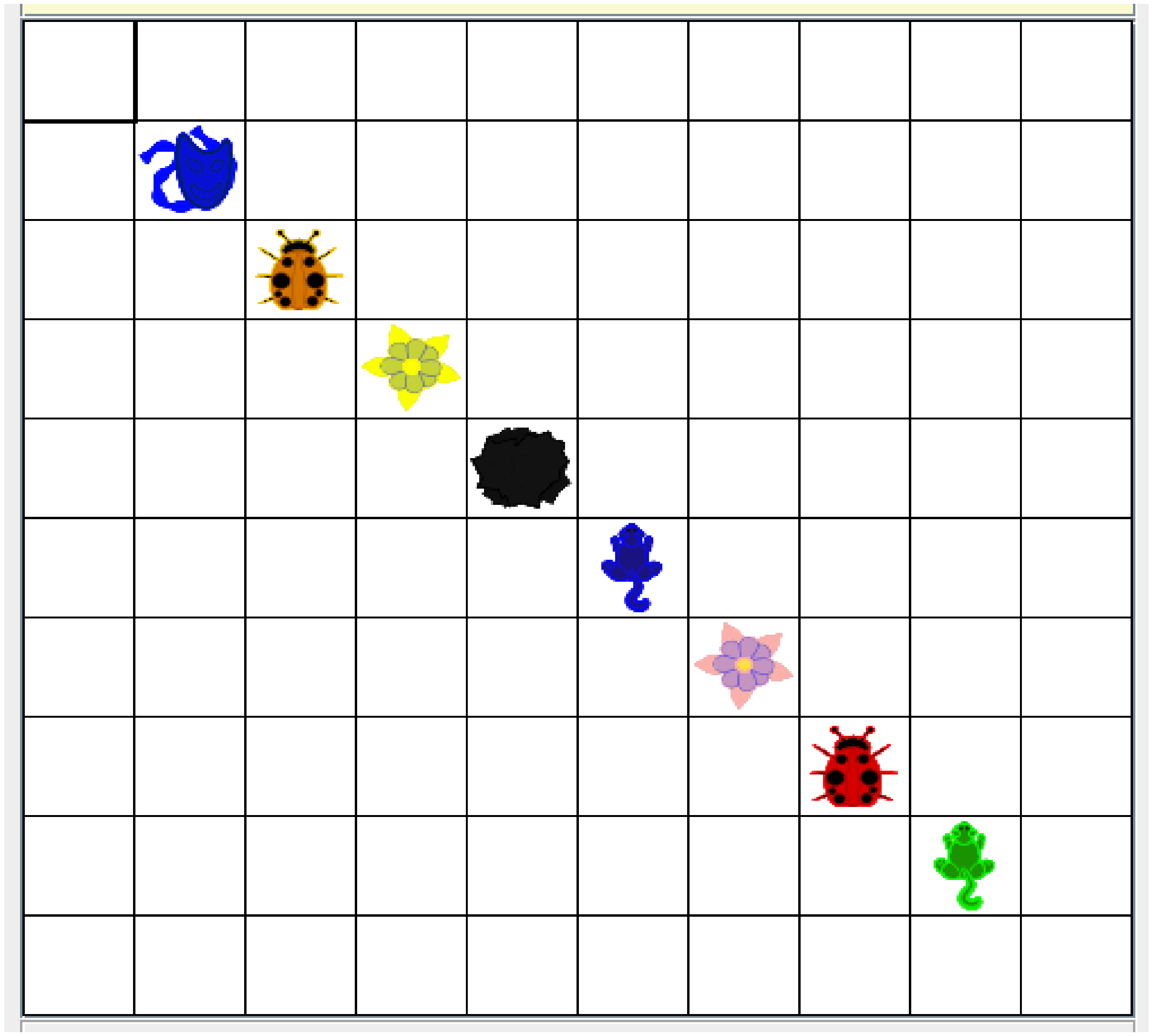


Click on a grid location to construct or manipulate an occupant.

Row = 2

Column = 1

What is GridWorld?



Grid

Grid

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
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

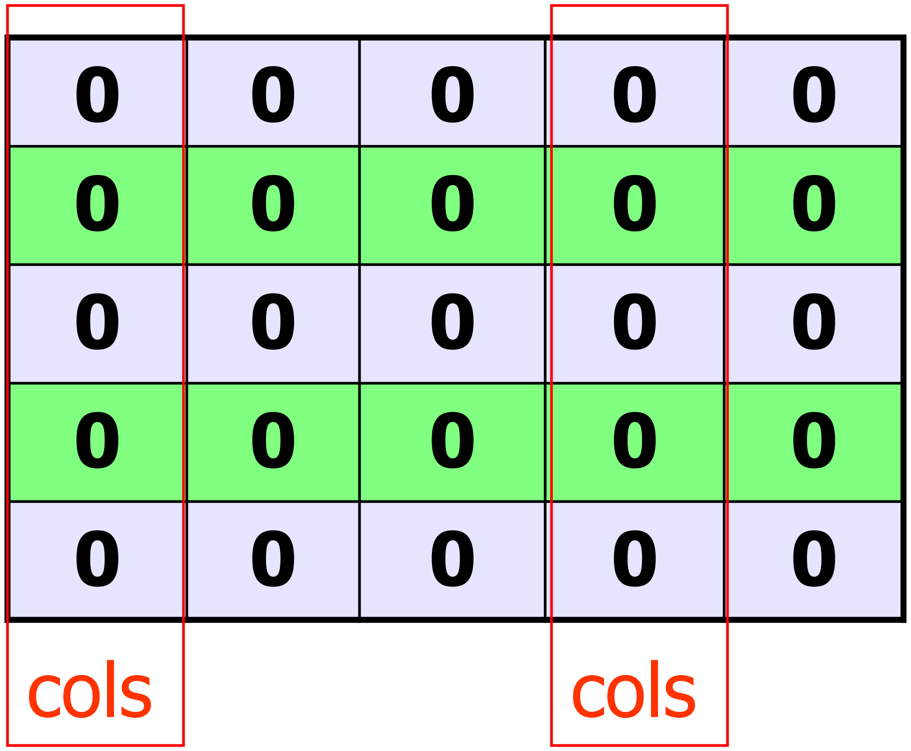
```
import info.gridworld.grid.Grid;
```

Grid

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.

Grid



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

A grid is a structure that has rows and columns.

Bug

Bug

Bug differs from actor in that a bug actually moves from cell to cell.

A bug moves to the cell immediately in front if possible. If a move is not possible, the bug turns in 45 degree increments until it finds a spot to which it can move.

Bug

extends Actor

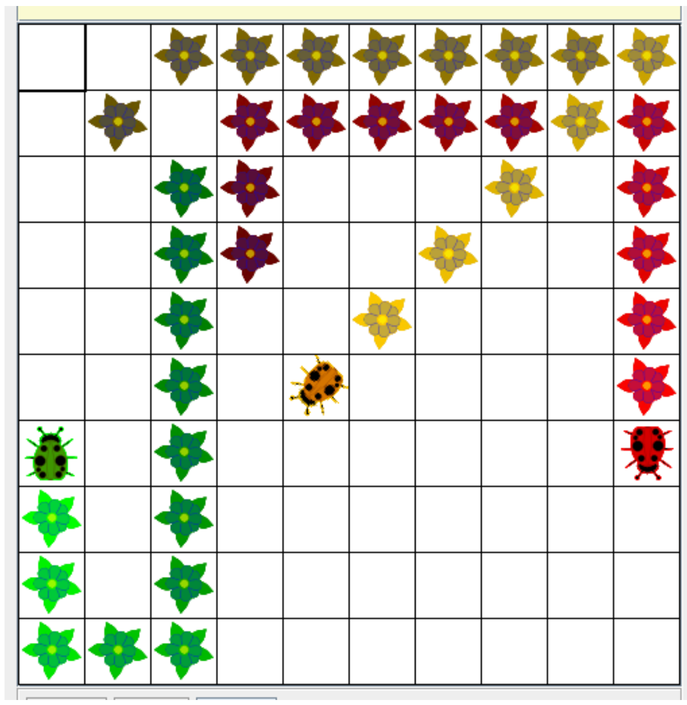
frequently used methods

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

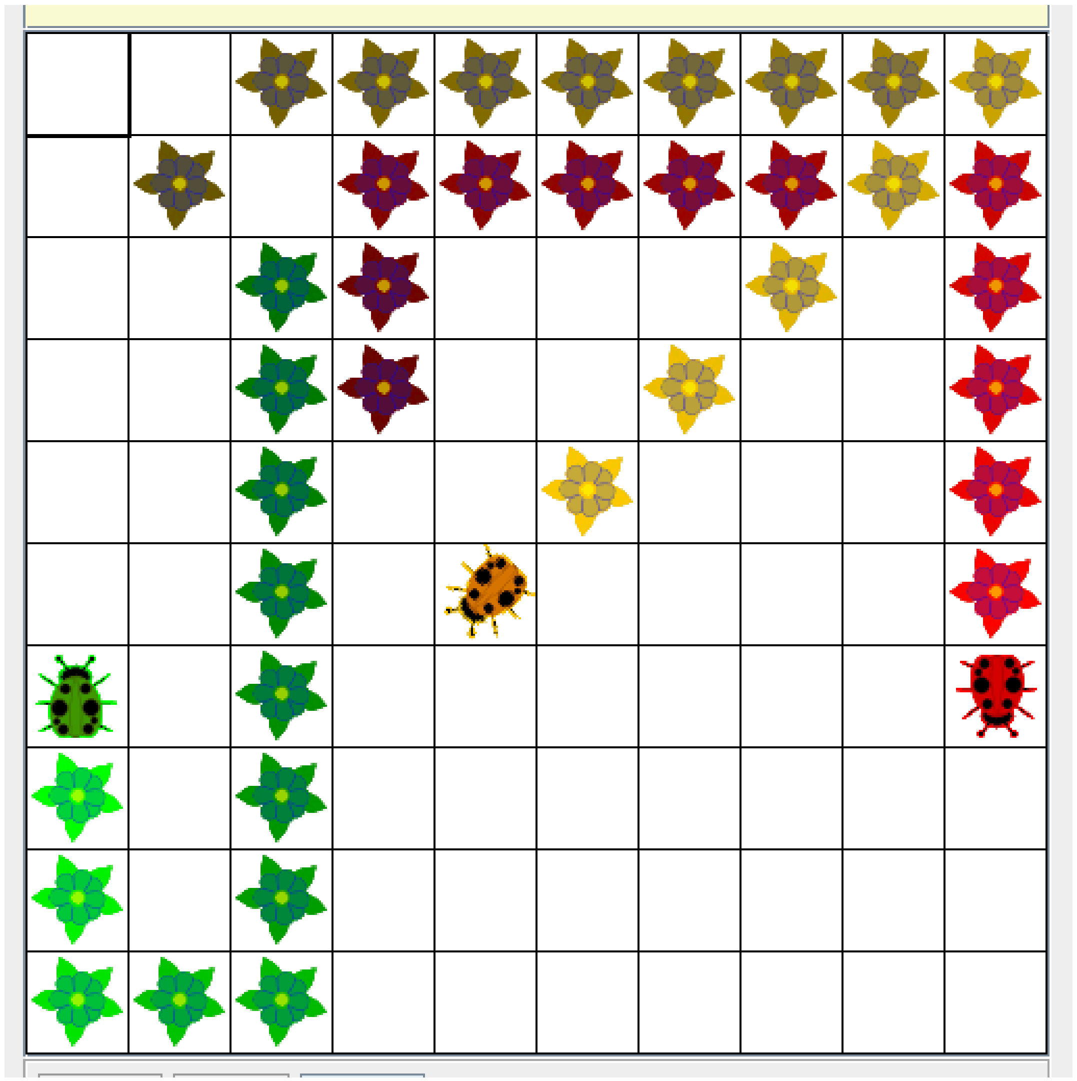
```
import info.gridworld.actor.Bug;
```

Bug

```
ActorWorld world = new ActorWorld();

Bug dude = new Bug();
world.add(new Location(3,3), dude);

Bug sally = new Bug(Color.GREEN);
sally.setDirection(Location.SOUTHEAST);
world.add(new Location(2,2), sally);

Bug ali = new Bug(Color.ORANGE);
ali.setDirection(Location.NORTHEAST);
world.add(new Location(1,1), ali);

world.show();
```



Bug



open
bugone.java

Bug

extends Actor

frequently used constructors

Name	Use
Bug()	make a new red bug going north
Bug(color)	make a new bug set to color

```
import info.gridworld.actor.Bug;
```

Bug

extends Actor

frequently used methods – Bug specific

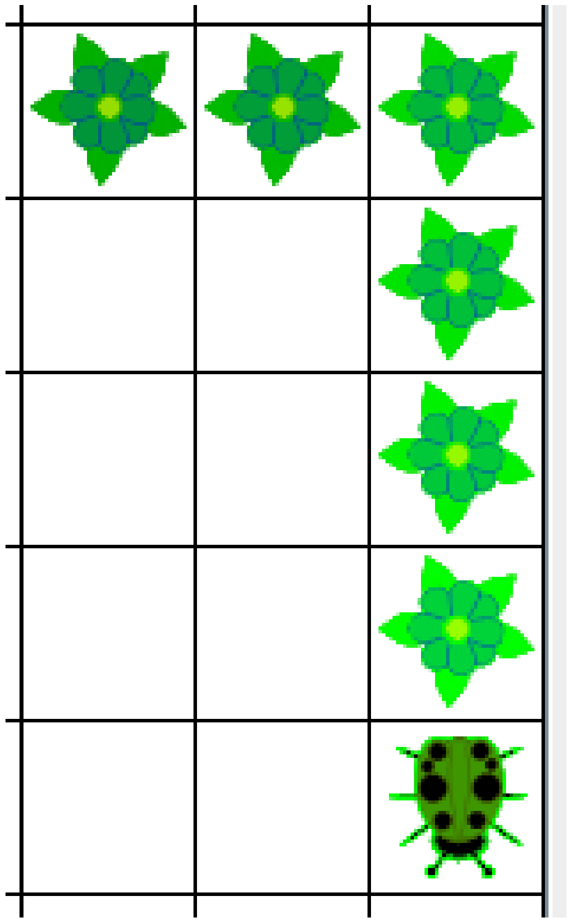
Name	Use
act()	move if possible or turn
canMove()	check to see if a move is possible
move()	move forward and leave a flower
turn()	turn 45 degrees without moving

```
import info.gridworld.actor.Bug;
```

Bug

What does a Bug do when its act() method is called ?

What methods does the act() method appear to call?

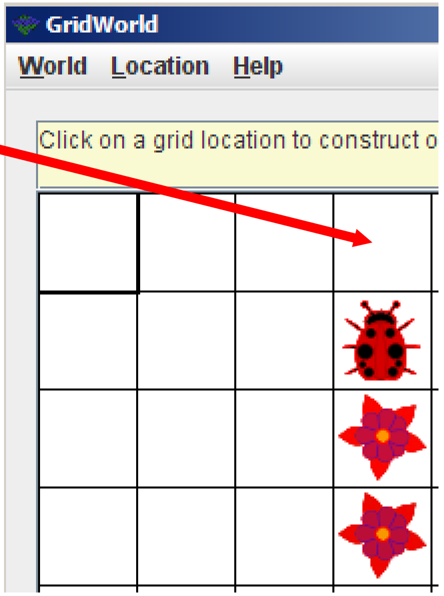


canMove

The bug act method looks to see if a move is possible by calling canMove.

canMove looks at the location in front of this bug to see if it is empty or if it contains a flower.

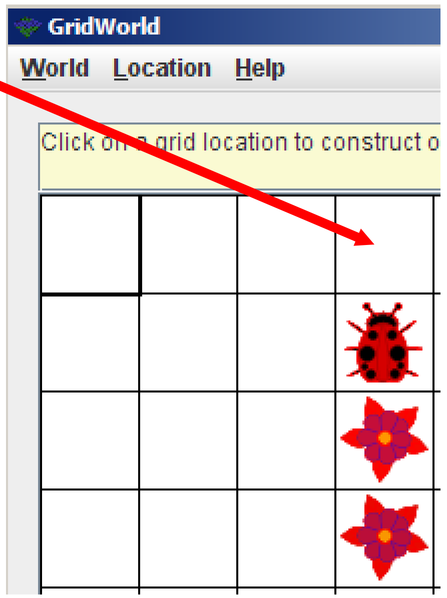
canMove returns true or false.



move

The bug act method calls move if canMove returns true.

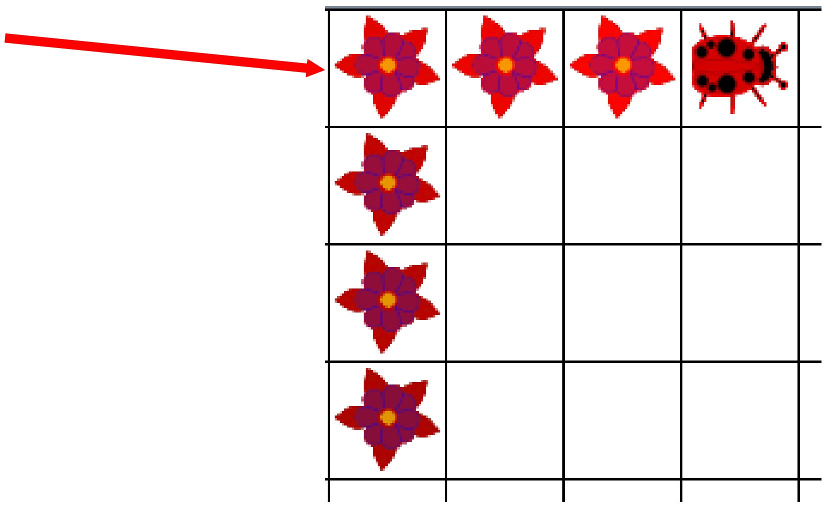
move calls moveTo to move the bug to the location in front of this bug.
move leaves a flower in the old location.



turn

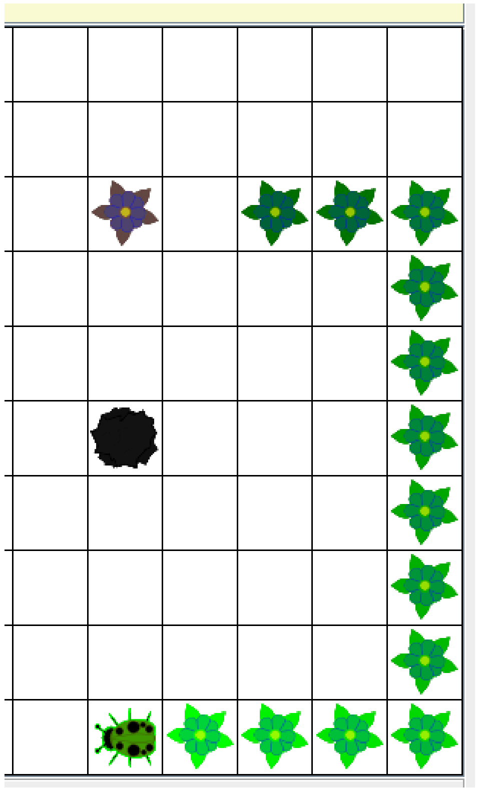
The bug act method calls turn if canMove returns false.

turn changes the direction of the bug by 45 degrees to the right.



Bug

```
ActorWorld world = new ActorWorld();
Bug dude = new Bug(Color.GREEN);
dude.setDirection(Location.EAST);
Location loc = new Location(5,5);
world.add(loc , new Rock());
loc = new Location(2,5);
world.add(loc, new Flower());
loc = new Location(2,7);
world.add(loc, dude);
world.show();
```



open bugtwo.java

Extending Bug

Extending Bug

How will the new bug differ from the original bug?

Can the new behavior be created using existing methods?

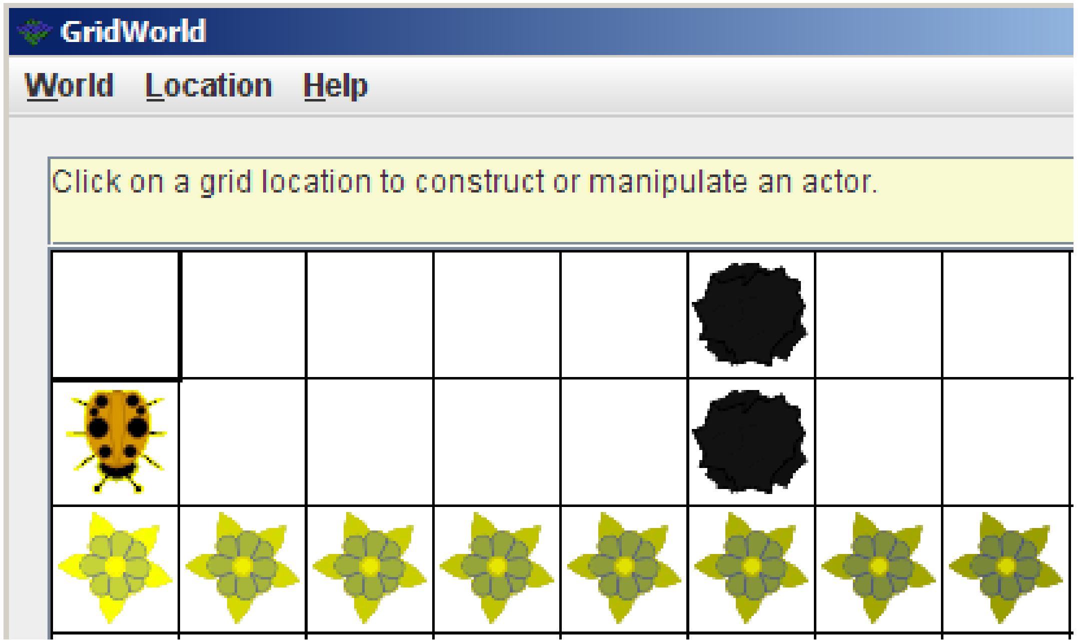
Which of the methods will be overridden?

Will new methods need to be added?

Extending Bug

What has to change if you want the bug to go backwards instead of forwards?

Use the GW quick reference!



Click on a grid location to construct or manipulate an actor.

Extending Bug

```
public class BackwardBug extends Bug
{
   //constructor

   public void act()
   {



   }

   //other methods

}
```

Is this the only way to write this class?

What methods could be changed?

open

backwardbug.java

backwardbugrunner.java

AP Exam Info

You will be given Bug and BoxBug in the quick reference when taking the AP exam.

You CAN override any of the BUG methods when making a new Bug.

Move and CanMove provide great examples of how to use getAdjacentLocation(), isValid, and get().

Always look at the original Bug and BoxBug code when making a new Bug.

AP Exam Info

You will be given Bug and BoxBug in the quick reference when taking the AP exam.

BoxBug also provides examples of instance variables, method overriding, and constructors.

You CAN override any of the BUG methods when making a new Bug.

Always look at the original Bug and BoxBug code when making a new Bug.

open

boxbug.java

boxbugrunner.java

Start work on Bug Exercises and Labs