Computer Science II
Gridworld Code
Paul L. Bailey

This is the source code for Gridworld which is supplied on the AP Exam.

Bug Class

```
package info.gridworld.actor;
import info.gridworld.grid.Grid;
import info.gridworld.grid.Location;
import java.awt.Color;

public class Bug extends Actor
{
    public Bug()
    {
        setColor(Color.RED);
    }

    public Bug(Color bugColor)
    {
        setColor(bugColor);
    }

    public void act()
    {
        if (canMove()) move();
        else turn();
    }

    public void turn()
    {
        setDirection(getDirection() + Location.HALF_RIGHT);
    }

    public void move()
    {
        Grid<Actor> gr = getGrid();
        if (gr == null) return;
        Location loc = getLocation();
        Location next = loc.getAdjacentLocation(getDirection());
        if (gr.isValid(next)) moveTo(next);
        else removeSelfFromGrid();
        Flower flower = new Flower(getColor());
        flower.putSelfInGrid(gr, loc);
    }

    public boolean canMove()
    {
        Grid<Actor> gr = getGrid();
        if (gr == null) return false;
        Location loc = getLocation();
        Location next = loc.getAdjacentLocation(getDirection());
        if (!gr.isValid(next)) return false;
        Actor neighbor = gr.get(next);
        return (neighbor == null) || (neighbor instanceof Flower);
    }
}
```

BoxBug Class

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

public class BoxBug extends Bug
{
    private int steps;
    private int sideLength;

    public BoxBug(int length)
    {
        steps = 0;
        sideLength = length;
    }

    public void act()
    {
        if (steps < sideLength && canMove())
        {
            move();
            steps++;
        }
        else
        {
            turn();
            turn();
            steps = 0;
        }
    }
}
```

Critter Class

```
package info.gridworld.actor;
import info.gridworld.grid.Location;
import java.util.ArrayList;

public class Critter extends Actor
{
    public void act()
    {
        if (getGrid() == null)
            return;
        ArrayList<Actor> actors = getActors();
        processActors(actors);
        ArrayList<Location> moveLocs = getMoveLocations();
        Location loc = selectMoveLocation(moveLocs);
        makeMove(loc);
    }

    public ArrayList<Actor> getActors()
    {
        return getGrid().getNeighbors(getLocation());
    }

    public void processActors(ArrayList<Actor> actors)
    {
        for (Actor a : actors)
        {
            if (!(a instanceof Rock) && !(a instanceof Critter))
                a.removeSelfFromGrid();
        }
    }

    public ArrayList<Location> getMoveLocations()
    {
        return getGrid().getEmptyAdjacentLocations(getLocation());
    }

    public Location selectMoveLocation(ArrayList<Location> locs)
    {
        int n = locs.size();
        if (n == 0)
            return getLocation();
        int r = (int) (Math.random() * n);
        return locs.get(r);
    }

    public void makeMove(Location loc)
    {
        if (loc == null)
            removeSelfFromGrid();
        else
            moveTo(loc);
    }
}
```

ChameleonCritter Class

```
import info.gridworld.actor.Actor;
import info.gridworld.actor.Critter;
import info.gridworld.grid.Location;
import java.util.ArrayList;

public class ChameleonCritter extends Critter

    public void processActors(ArrayList<Actor> actors)
    {
        int n = actors.size();
        if (n == 0)
            return;
        int r = (int) (Math.random() * n);

        Actor other = actors.get(r);
        setColor(other.getColor());
    }


    public void makeMove(Location loc)
    {
        setDirection(getLocation().getDirectionToward(loc));
        super.makeMove(loc);
    }
}
```