DEADLY DOSE

New Mexico
Supercomputing Challenge
Final Report
April 4, 2012

Team 106
Rio Rancho Cyber Academy

Team Members:

Sierra Venegas

Jocelyn Tansey

Monika Nadzins

Teacher:

Harry Henderson

Summary

Radiation. To some, this can be a deadly thing, not to be played with. To others, it’s just another word in the dictionary. The agent based simulation, “Deadly Dose” shows how radiation can effect a community in whole. With the use of NetLogo, a program was devised to show people how the radioactive element of radon, used in many nuclear power plants and labs, can effect their health.

“Deadly Dose” has two types of people, workers and citizens. The workers could go in and out of the nuclear lab in the center of the simulation as they wished. Citizens, however, were restricted from access to the lab itself. The workers spread most of the initial radiation. After the radiation left the lab, citizens were exposed. Along with the spread of radon, came overexposure for some unfortunate people. The overexposure ultimately affected the workers, who were almost constantly exposed. The abundance of radon caused death for many.

The program has many different options for user interaction. These factors on the interface include choosing a startup amount of workers and citizens, a slider to choose the startup energy level and the maximum population. The “Deadly Dose” simulation has many realistic factors, such as energy, reproduction, death, water resources, and workers that retire and are hired. All these factors play a major role in a community in real life situations, and therefore should be in a simulation.

Problem

High exposure to radiation is a hazard to many, especially in recent years. Nuclear plants all around the world have been established, producing high amounts of radioactivity and threatening the lives who come in contact with it. Scientists have discovered multiple ways in which this dangerous substance can impact human health. Gastro vascular and neurovascular abnormalities have been observed in some unfortunate exposures. From gamma to beta, rays of this deadly element have been known to affect the anatomy of a healthy human being. In our project, we tackled and described these main effects and produced a simulation that can illustrate the odds of radiation consumption as well as who it affected, and why.

Methods

“Deadly Dose” is an agent based simulation that was programmed under NetLogo. The program uses real life variables. Many of these variables include two types of people. Workers and citizens. The workers have unlimited access to anywhere in the community, including the lab. Although this may seem like a privilege to some, it has much more risks. The citizens, on the other hand, can go anywhere in the community outside the lab border. This is to show that not everybody works in radioactivity in a community.

There are many more features other than the people. The people themselves have one mandatory variable, radiation, and one optional variable, energy. The energy factor of the program is mainly to allow the people to reproduce, die, and move. Just like regular humans, the virtual people in the program can lose and gain energy. When the program starts, you can decide what energy level to start the population off with. It is preferably set at 300. As the people move forward one space, they lose five energy points. They can gain energy back however, by going to water resources.

Water resources are available to anyone. They are virtual water wells (blue squares) that, when stepped on, can give the individual 900 energy points. If a person loses all of their energy, they will die. They also die if they have to much energy. As unusual as it may sound, this actually controls the population. If someone has more than 25,000 energy points, they die. When this is compared to real life situations, it can be recognized as a stress induced heart attack or stroke.

Radiation acts in a similar matter. If an individual person receives more than 3,500 radiation, they will die. This is based on the amount of radon it takes to kill somebody with a neurovascular affect. People gain radiation by stepping on a radiated area. Radiated areas are recognized by the color of the area. If it is white, then it is more densely radiated, and the more radiation they gain. They can also gain additional radiation from water resources that have been exposed to the radon. The darker the water, the more radiated it is.

When the people become radiated, you will notice that they may change. Citizens turn orange in color and gain a radiation amount. Workers turn red. The program allows the user to choose if they want the radiation amount to be labeled above the person's head or not. If any given person has a certain amount of radiation, they spread the radiation to the area that they are located.

NetLogo is a great agent based program to design a simulation such as Deadly Dose. It allows the user to setup the program how they wish and watch the community grow, and yes, die. This model is great to show how radon can affect certain communities.

Models

The Interface:

Figure 1.

The initial s

tartup view of the community

Figure 2.

Total populations promptly after starting program.

Figure 3.

Amount of radiated land and resources promptly after starting the program.

Figure 4.

Amount of radiated workers (red) and general workers (green) promptly after starting the program.

Figure 5.

Amount of radiated citizens (orange) and general citizens (blue) promptly after starting the program.

Code: (NetLogo)

```
breed [workers worker]

breed [citizens citizen]


turtles-own [radiation
```

```
                    Energy]
globals [halfedge]


to go
   move-turtles
   infect
   radiate-resources
   live
   check-death
   reproduce
   hire
   retire
   graph-totals
   graph-workers
   graph-citizens
   graph-radiated-land
end

to setup
   clear-all        ;; clears everything
```

```
setup-source
setup-resources
create-workers startup-workers    ;; creates workers according to slider
[
    if (include-energy? = "yes")    ;; this tells the workers to use energy as a variable
    [                                                      ;; when the drop down menu is put to "yes".
        set energy energy-level          ;; they get the energy level according to the slider
    ]
    set breed workers
    set energy energy-level
    set color green
    setxy random-xcor random-ycor
    set shape "person construction" ;; sets shape of workers to green construction people
]
create-citizens startup-citizens ;; creates citizens according to slider
[
    if (include-energy? = "yes") ;; same chooser as the workers'
    [
        set energy energy-level
    ]
    set breed citizens
```

```
      set color blue

      placement

      set shape "person"           ;; sets shape of turtles to blue people

   ]

end


to placement

   setxy random-xcor random-ycor

   if xcor ^ 2 + ycor ^ 2 <= (edge * .75) ^ 2 ;; this tells the citizens that they cannot

   [                                                                  ;; be in the border of the power plant
when

      placement                                                      ;; the program is set up.

   ]                                                                  ;; you can adjust the size of the border

end                                                                    ;; with the "edge" slider


to move-turtles

   ask turtles                 ;; this asks all of the turtles in the world to move

   [

      right random 30

      left random 30           ;; moves turtles left/right randomly

      ifelse color = green or color = red ;; only workers listen to this command

      [
      forward 1    ;; moves workers forward 1

      set energy energy - 5 ;; they lose 5 energy with every step

    ]

 ; else they are blue/orange citizens

    [

      ifelse xcor ^ 2 + ycor ^ 2 <= (edge * .75) ^ 2 ;; citizens cannot pass the power

      [                                              ;; plant border

        rt 180                                      ;; so they turn around and
begin walking

        fd 1

      ]

      [

        forward 1 ;; moves citizens forward 1

        set energy energy - 5 ;; they lose 5 energy like the workers

      ]

    ]

  ]

  ask citizens

  [

    if radiation < 900

    [if radiation > 5

     [ if pcolor < 9
```

```
          [set pcolor pcolor + .1]
        ]
      ]
  ]
  ask workers
  [
      if radiation < 900
      [if radiation > 5
        [ if pcolor < 9
            [set pcolor pcolor + .1]
        ]
      ]
  ]

end

to live              ;; this tells all of the turtles that they can gain energy to continue living
  ask turtles ;; with water.
  [
      if (include-energy? = "yes")
      [
     if pcolor = 85            ;; this tells all workers and citizens that if they step on
     [                         ;; blue water
      set energy energy + 900 ;; they gain 900 energy
     ]                         ;; but only if the "include-energy?" chooser is set to
"yes"
    ]
  ]
  ask workers
  [
    if radiation <= 0
    [ set energy energy + 90 ]
  ]
  ask citizens
  [
    if radiation <= 0
    [ set energy energy + 90 ]
  ]

end

to infect   ;; this is the command that spreads the radiation
 ask workers
```

```
[
  if radiation >= 1
  [ if pcolor >= 0 and pcolor < 10
   [
    set color red
    set radiation radiation + (pcolor * 10)
    set pcolor pcolor + .02
   ]
  ]
]

 ask workers
   [
     ifelse show-radiation?     ;; this is for the switch to show the level of radiation
      [ set label radiation ]  ;; above each worker's head
      [ set label ""]
   ]
 ask citizens
 [
   if radiation >= 1
   [
            if pcolor >= 0 and pcolor < 10

            [

            set color orange

            set radiation radiation + (pcolor * 10)

            set pcolor pcolor + .02

        ]

    ]

    ask citizens

    [

      ifelse show-radiation?    ;; this is the same switch from the workers

      [ set label radiation ]

      [ set label ""]

    ]

    ]

end




to setup-source          ;; makes the power plant

    repeat 1
```

```
  [
    ask patch 0 0    ;; puts the power plant in the center
    [
      set pcolor 9 ;; and turns it white
    ]
    diffuse pcolor .2        ;; tells center patch to be white and to diffuse (the radiation)
  ]                                        ;; the source is diffused by .2
end


to setup-resources ;; this command sets up resources such as water.
  ask patch -9 1
  [
    set pcolor 85 ;; this is telling patch (-9, 1) to turn 85
    ask neighbors ;; and tells the immediate surrounding patches to do the same
    [                          ;; this is water
      set pcolor 85
    ]
  ]
  ask patch 1 -9
  [
    set pcolor 85
    ask neighbors
    [
        set pcolor 85
    ]
]
ask patch 11 13
[
    set pcolor 85
    ask neighbors
    [
        set pcolor 85
    ]
]
ask patch -13 11
[
    set pcolor 85
    ask neighbors
    [
        set pcolor 85
    ]
        ]
```

```
end


to radiate-resources    ;; this command tells the water that if it gains radiation from a person
    ask turtles                  ;; to turn a darker color to show that the water is radiated
      [
        if radiation >= 9
        [
            if pcolor <= 85
            [if pcolor >= 82
            [ set pcolor pcolor - .1
                set radiation radiation - 15]
            ]
        ]
    ask citizens      ;; citizens get radiated from water that is radiated...
    [
      if pcolor <= 84.5
        [
            if pcolor >= 82
          [
              set radiation radiation + 5
              set color orange ;;... and turn orange and gain 5 radiation
          ]

        ]

    ]

    ask workers     ;; workers get radiated from the water that is radiated like the citizens...

    [

        if pcolor <= 84.5

        [

            if pcolor >= 82

            [

                set radiation radiation + 5

                set color red ;;... only difference is that they turn red.

            ]

        ]

    ]

    ]


end


to restrict          ;; draws imaginary square border for power plant in center of world

    set halfedge int (edge / 2)

    ask patches [
    if (pxcor = (- halfedge) and pycor >= (- halfedge) and pycor <= (0 + halfedge) )
      [set pcolor magenta]
    if ( pxcor = (0 + halfedge) and pycor >= (- halfedge) and pycor <= (0 + halfedge) )
      [set pcolor magenta]
    if ( pycor = (- halfedge) and pxcor >= (- halfedge) and pxcor <= (0 + halfedge) )
      [set pcolor magenta]
    if ( pycor = (0 + halfedge) and pxcor >= (- halfedge) and pxcor <= (0 + halfedge) )
      [set pcolor magenta]
    ]     ;; it does not turn the barrier magenta because the magenta would diffuse along with the white radiated patches
end


to reproduce ;; this command tells turtles to reproduce
  ask turtles [
   if count turtles < maximum-population[
    if energy >= 4000
    [if energy < 25000
    [
     hatch 1
      [ set energy 500
        set radiation radiation / 2]
      set energy 0
```

```
        ]
       ]
       ]
     ]



end


to hire                                          ;; when a worker dies or retires a new one is hired
    ask turtles
          [ if breed != workers [
              if count workers < 40
                  [set breed workers
                    set radiation 0
                    set shape "person construction"
                    set color green
                    set energy energy + 900]
              ]
          ]

    ask turtles [
              if pcolor > black and pcolor < 9[
                  infect
          ]
    ]



end


to retire                                  ;; tells workers to retire (become citizens) if they lose so much energy
    ask workers
    [ if energy <= 2000
        [
            set breed citizens
            set shape "person"
            set color blue
            set energy energy + 900
            set radiation radiation / 4
        ]
    ]
end
```

```
to graph-totals ;; this command is to make a plot for all of the people and radiation

    set-current-plot "Totals"

    set-current-plot-pen "citizens"

    plot count turtles with [color = blue] ;; plots for non-radiated citizens

    set-current-plot-pen "workers"

    plot count turtles with [color = green] ;; plots for non-radiated workers

    set-current-plot-pen "radiated-citizens"

    plot count turtles with [color = orange] ;; plots for radiated citizens

    set-current-plot-pen "radiated-workers"

    plot count turtles with [color = red] ;; plots for radiated workers

    set-current-plot-pen "radiation"

    plot count patches with [pcolor <= 9 and pcolor >= .4 and pcolor <= 84.5 and pcolor >= 82] ;;
plots for radiated patches

end


to graph-workers ;; this is a graph for only workers

    set-current-plot "Workers"

    set-current-plot-pen "workers"

    plot count turtles with [color = green]

    set-current-plot-pen "radiated-workers"

    plot count turtles with [color = red]

end
```

```
to graph-citizens ;; this is a graph for only citizens
    set-current-plot "Citizens"
    set-current-plot-pen "citizens"
    plot count turtles with [color = blue]
    set-current-plot-pen "radiated-citizens"
    plot count turtles with [color = orange]
end

to graph-radiated-land
    set-current-plot "Radiated Land"
    set-current-plot-pen "radiated-land"
    plot count patches with [pcolor <= 9 and pcolor >= .4]
    set-current-plot-pen "radiated water"
    plot count patches with [pcolor <= 84.5 and pcolor >= 82]
    set-current-plot-pen "non-radiated area"
    plot count patches with [pcolor < .4 and pcolor = 85]
    set-current-plot-pen "total radiated area"
    plot count patches with [pcolor <= 9 and pcolor >= .4]
    plot count patches with [pcolor <= 84.5 and pcolor >= 82]
end
```

```
to check-death ;; this command tells turtles to die
    ask turtles
    [
        if radiation >= 3500 [ die ] ;; they can die if their radiation is 3500 or more
        if energy >= 25000 [ die ] ;; or if energy is 25,000 or more (stress induced strokes/heart attacks for population control)
    ]
end
```

Results

After much observance of the program, the community was sustainable population wise. As some people did in fact die, there were many that lived. The radon was passed down to new generations, but soon reduced its potency, as if it had a virtual half-life.

Also observed, was the fact that workers did die much faster than the citizens. This variable did show that overexposure to radon can kill you. The workers were allowed to go inside the lab at any time, while the citizens were restricted from it. Since the citizens were restricted, the workers were the people who spread the initial radiation. After the radiation was exposed to the outer community, citizens could get radiated.

The community remained sustainable with the workers and the citizens. As soon as a worker died, a new worker was “hired”. This meant that, when a worker died (of either radiation or low energy), the program would ask a random citizen (if they had so much energy) to turn into a worker. This program showed actual scenarios of labs that use the element radon.

Although the results were not as expected, they were very informative. They showed that a population can remain sustainable, but will still receive an unusual death rate due to the radiation. It also showed that radiation spreads faster than we wish to believe. True, there is a half-life that is unique to each radioactive element, but we can still use this model to teach the public how to protect themselves if they are in a similar situation.

Achievements

“Deadly Dose” had many great achievements. One of the most significant, was virtually proving that overexposure to certain elements, in this case radon, can kill you. People who are surrounded by this, or other, radioactive elements should take extra precautions as to what they do. Even though, it may seem like it’s no big deal, radiation can kill. Especially if you get a “Deadly Dose”.

The program also shows how a population can remain sustainable with a radioactive element such as radon, but will also be susceptible to its major affects. The program can be used to educate others on how radiation can affect them, especially when it’s in their own backyards.

Of course, like other models, “Deadly Dose” can be altered. Our team is planning on extending on this model next year to gain further knowledge of the deadly elements that may be just down the street. “Deadly Dose” was programmed under NetLogo to help show people how nuclear labs work, not make them solve it. This model has truly achieved it’s potential for this year, with hopes of raising the bar next year.

Acknowledgements