Modeling an Ecosystem

New Mexico
Supercomputing Project
Final Report
April 2nd 2008
Team 5 Albuquerque Academy

Team Members:
David Baack
Nick Longenbaugh

Teacher
Jim Mims

Mentor
Kevin Fowler

Table of Contents

Executive Summary

In the ideal ecosystem, the plant and animal species live together harmoniously; the predator and prey populations oscillate as more prey are eaten or as predators starve, and the resources are consumed and regenerated, but on the whole the ecosystem is static. Once again, the ecosystems fluctuate constantly, but at the same time, they remain stable overall as organisms evolve to adapt to the changing environment. In this natural state, ecosystems are a perfect balance of growth and decay, predation and reproduction. However, the ideal simulation can effectively model very few, if any, real world ecosystems. The real world is not ideal. Natural and man made disasters can throw the balance between predators, prey, and resources out of sync. Our project allows the user to create an ideal ecosystem, then introduce different disasters and problems and test how the populations respond.

Statement of Problem

Our problem is threefold. First, we had to create a model that could effectively exhibit the behavior of an ideal ecosystem, namely one with one species of predator, one species of prey, and one resource. The prey consumes the resource, the predator consumes the prey, and the predator and prey reproduce. The resource also regenerates. Based on the initial conditions, namely resource quantity, prey populations, and predator population, as well as the rates of reproduction and resource regeneration; we had to create a model that demonstrated how the populations and resource quantities varied with time.

Once we created the model for this ideal ecosystem, we had to modify it to allow for disasters and other adverse situations. We had to see how such things as drought (reduced resource availability), human intervention in the form of hunting, pollution, and viruses effected the populations and resources.

Once we finished this second model, we had to determine just how severe of a disaster the ecosystem could sustain without having all of the species present go extinct. For example, how long can the ecosystem sustain a drought, or how severe can the air and groundwater pollution be before the species die? Our model was designed to answer these questions.

Method

We examined population time data for an ecosystem consisting of bighorn sheep and mountain lions. After seeing just how these populations varied with time we created a model that accurately predicted the populations of these species as a function of time. This basic model of the static equilibrium allowed the populations to flux, but neither species ever went extinct. First, we created a model with only sheep and grass. We divided the screen into a tiled area where the sheep "lived." Each of the tiles was either grass or dirt. The sheep moved randomly across the screen. Each sheep has a set amount of initial "life", and every time they move, they lose some life. If they encounter a square with grass, they regain a certain amount of life. If the sheep every run out of life, they disappear. Each sheep also has a chance to reproduce each turn. If they do reproduce, a new sheep is created. Once a square of grass has been consumed, it has a small chance to "re-grow." The user can control most of these variables to see how each effects the sheep population as a function of time. Once we finished this model, we added in a mountain lion population. The mountain lions move in the same manner as the sheep. However, they cannot gain life from eating grass. Instead, they must "eat" the sheep. Every time a mountain line runs across a square, which also contains a sheep, the sheep disappears and the mountain line gains a large amount of life. Also like sheep, mountain lions have a small chance of reproducing each turn. Now, we used published research data to modify the values of initial population, resource regeneration, reproduction rates and life span to create an equilibrium condition in which neither species ever goes extinct.

Once we created this model, we developed a series of "disasters" which the user can unleash upon the ecosystem to see what effect each disaster has upon the populations.

The first such disaster, adding a factory, has several effects on the population. First, each time a factory is added, the reproduction rates of the animals decrease. This is meant to model the effects pollution has on the reproductive health of animal species. In addition, each time an animal enters the same tile as the factory it dies. This is meant to model the fact that the land immediately surrounding the factory is often polluted to deadly levels and devoid of resources.

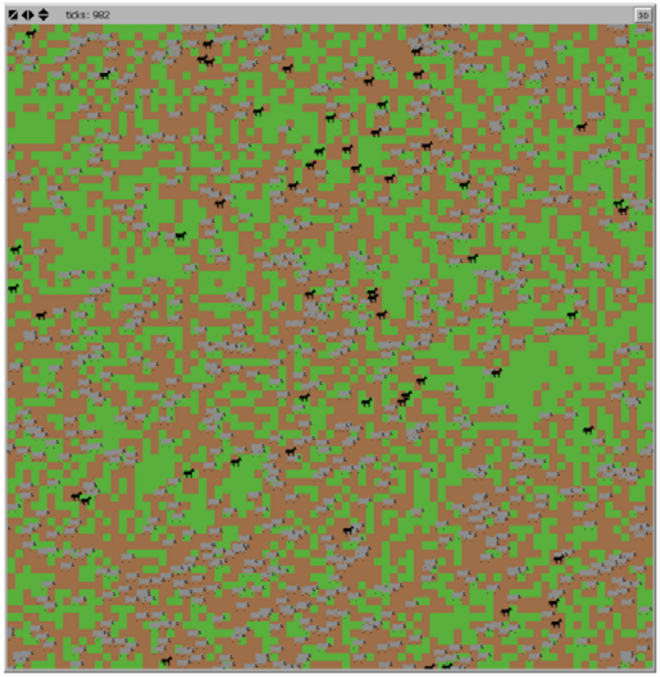
The next disaster is the introduction of hunting into the ecosystem. To model this, we added a new animal, people, to the project. The people move around in the same manner as the sheep and lions. However, each time a sheep or lion encounters a person, the person kills the animal. Thus, the people move around the screen killing the sheep and lions. People have a limited life, and they cannot gain health. Rather, they roam around for a short time killing, then disappear. This is meant to model the concept of hunting seasons; in short, people are only a part of the ecosystem for a very limited time.

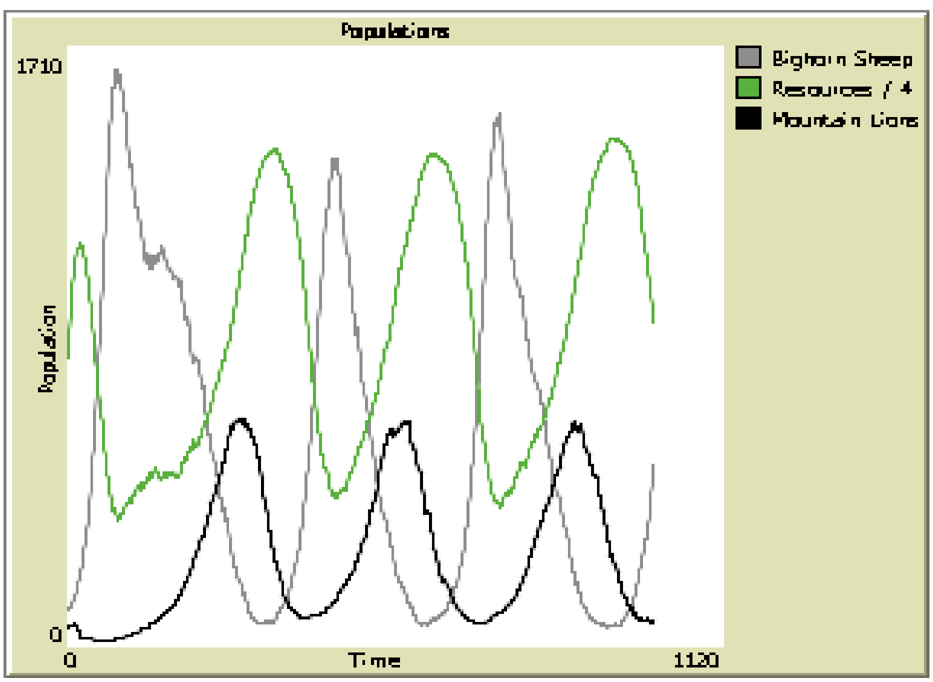
The next disaster is a drought. This simply reduces the rate at which the grass re-grows. During real droughts, limited rainfall stunts the growth of plants and limits available water supplies. Thus, we just made the grass grow at a slower rate to model how well species survive with limited resource availability.

The final disaster is the introduction or a virus into the ecosystem. The virus begins as an airborne entity that has a chance to infect every animal it encounters. If an animal does become infected, it looses more life with each tick. In addition, infected animals have a chance of infecting other animals they encounter. Thus, the virus spreads throughout the population and reduces the life span of infected animals.

Results

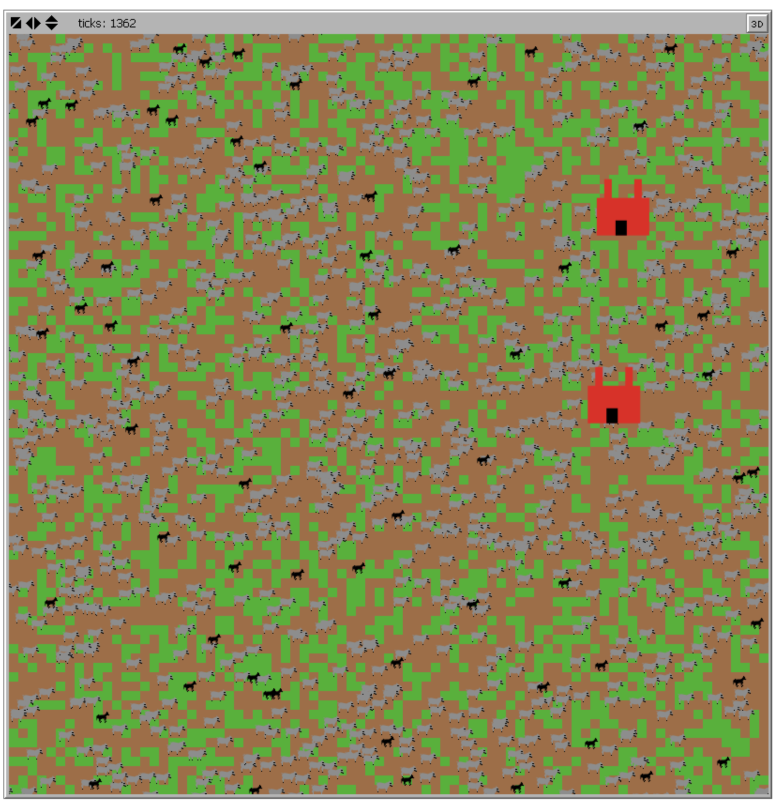
Equilibrium



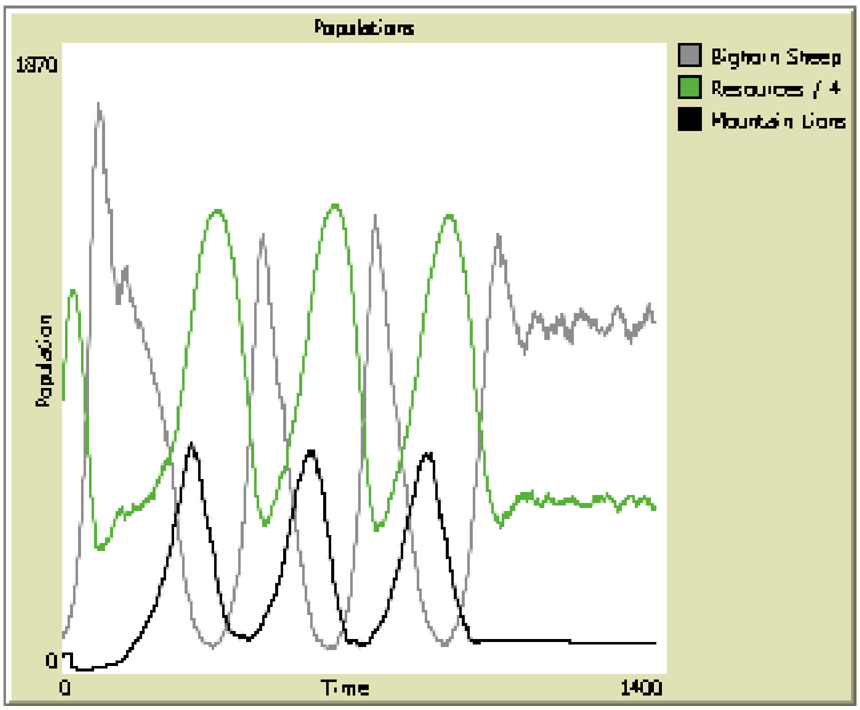


This is a graph of our population after running the simulation for 1000 ticks under normal settings. That is, these settings allow the ecosystem to achieve equilibrium. As you can tell from the graph, the population of sheep (represented by the gray line) and the population of lions (represented by the black line) oscillate, as they should according to a simple predator-prey model. As the population of one species reaches its highest point, the population of the other simultaneously reaches its lowest point. Then the low population rises and the high population falls until the two reverse roles, in a never-ending power struggle between the two species. What is important to note, however, is that neither species ever goes extinct, but always recovers after reaching its lowest point. Thus, the system is in equilibrium.

Factory

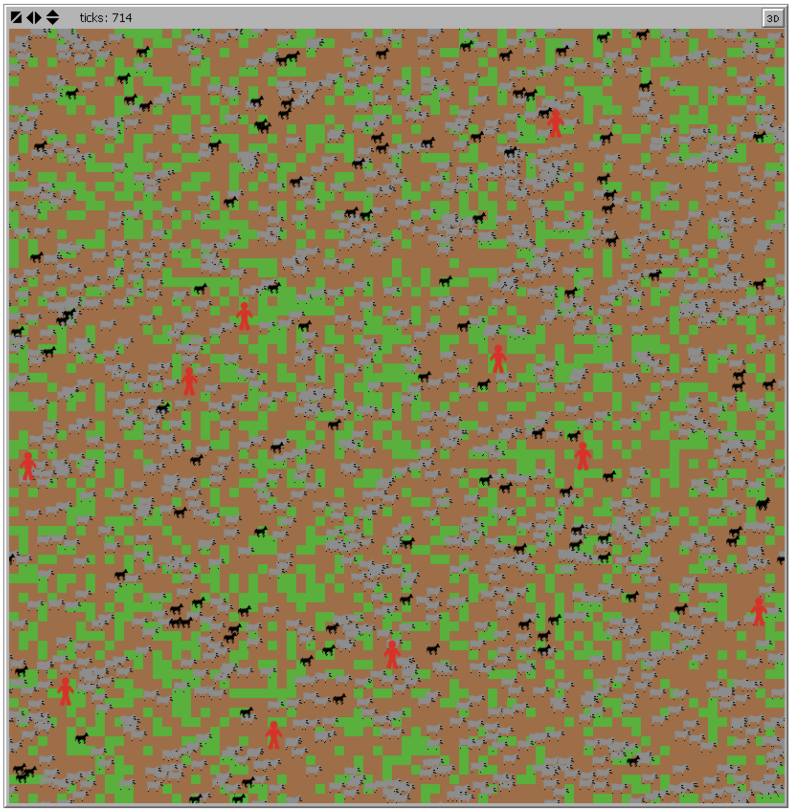


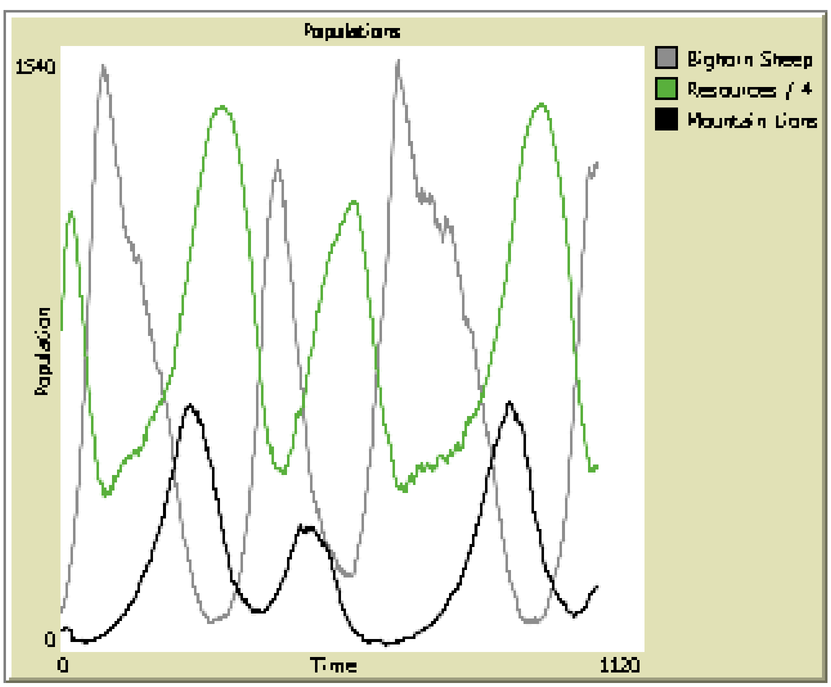
First, we ran the simulation until the two species established equilibrium with each other as before. Then we introduced two factories into the system, killing all animals in a certain radius of each factory and reducing the reproduction rates of all species. This is the result:



It's easy to tell the point at which the factories were introduced, because the lion population reached a minimum but did not recover as it does in equilibrium. Also, the sheep population did not decrease nearly as much as usual after its maximum, instead establishing a fairly stable population of about 1000, and maintaining it with very little oscillation. The lion population began to decrease very slowly, since they almost never reproduced and died more readily due to the factories' effect. If we let the simulation run for a much longer period of time, the lions eventually completely die out, but the sheep maintain a stable population of about 1000.

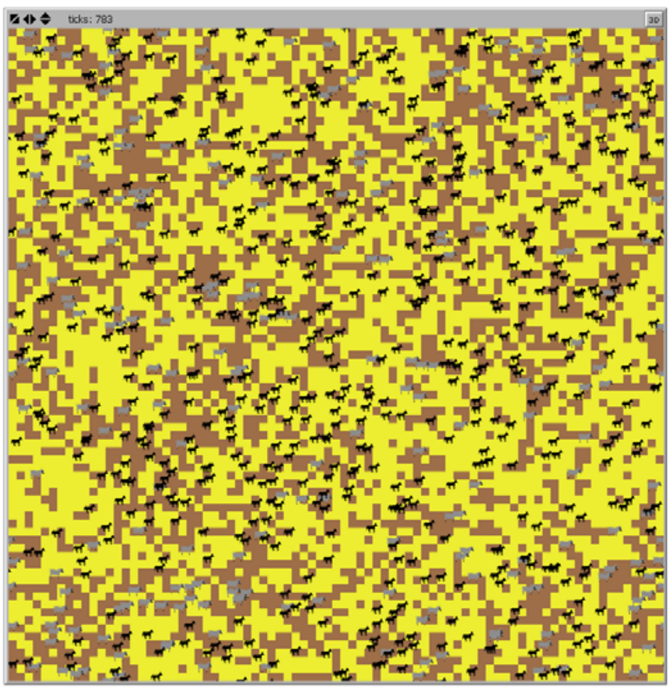
Hunters

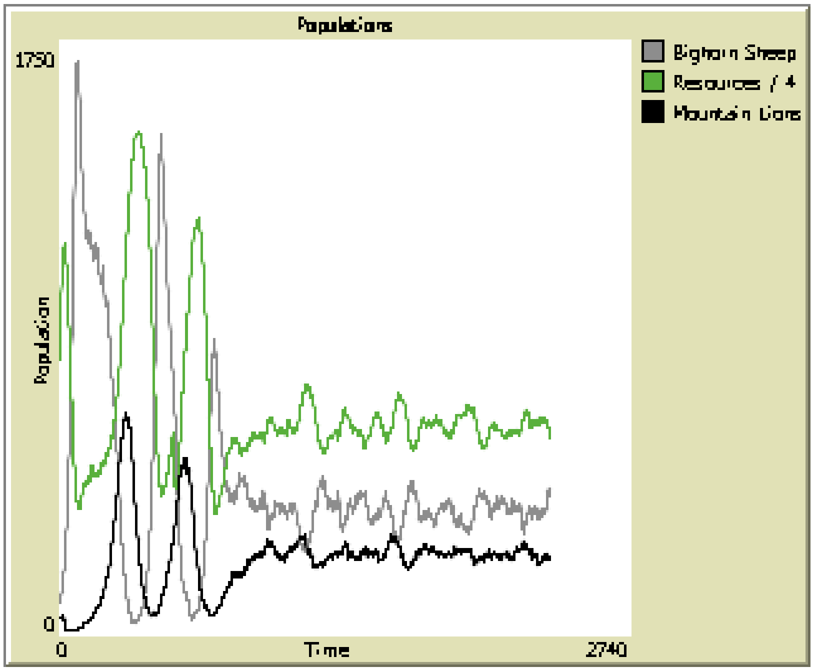

ticks: 714
3D


Populations
1540
0
Population
0
Time
1120
Bighorn Sheep
Resources / 4
Mountain Lions

This time we added hunters after the system reached equilibrium. You can tell because the lion population decreased by a great amount, nearly becoming extinct. At the same time, however, the sheep population reached an absolute maximum in spite of the presence of hunters. This is because there were almost no lions left to eat the sheep, and the humans that replaced them were not as much of a menace. Once the hunting season ended, the lion population bounced back and the ecosystem reached the exact same equilibrium it had before. Thus, hunters do not have any long-term effects on ecosystems as long as they do not hunt any species to extinction.

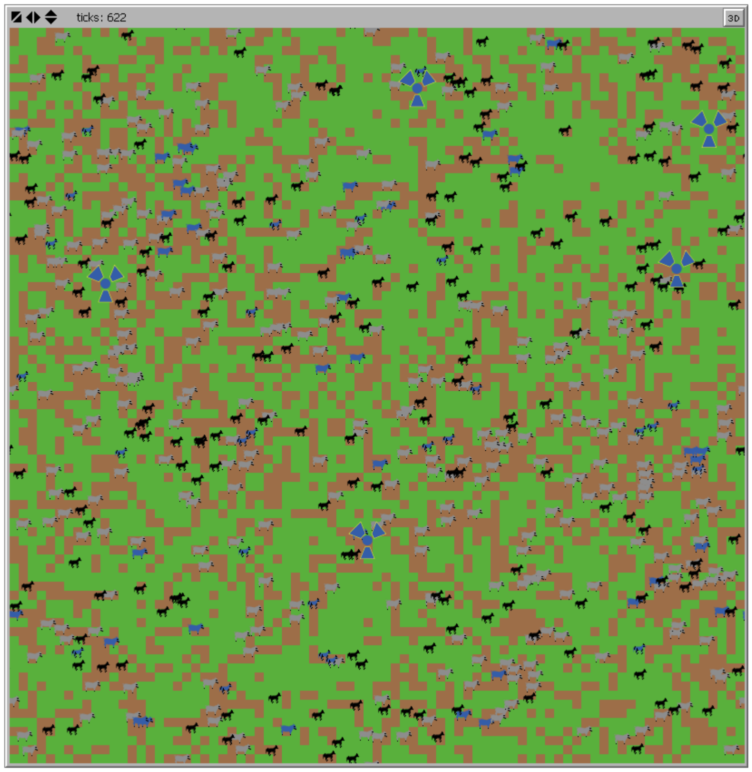
Drought

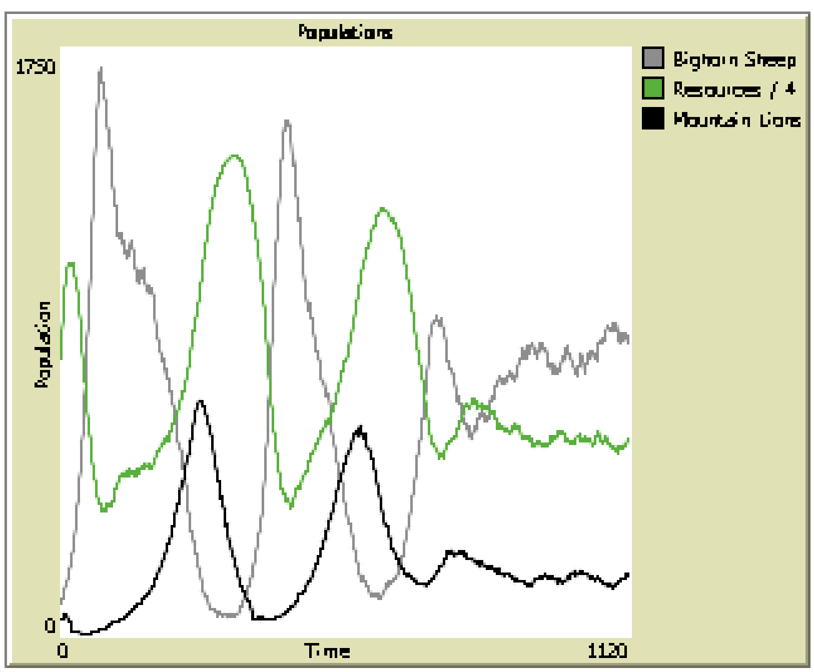




We introduced a drought into the system this time, reducing the rate at which recourses replenish. The effect this had is that each of the total populations decreased slightly, but more importantly the oscillations greatly decreased in amplitude, giving each species a more stable population size. We speculate that because of the reduced availability of resources, the sheep no longer had the food to reach such a high maximum population, and therefore the lion population couldn't reach as great a low. Thus, both populations oscillated less after the drought set in.

Virus


ticks: 622
3D


Populations
1750
0
Population
0
Time
1120
Bighorn Sheep
Resources / 4
Mountain Lions

The virus has the effect of reducing the lifespan of all infected animals, and has a chance of being passed on each time an infected animal encounters an uninfected one. After we introduced this virus into the system, the population of the sheep did not reach as high a maximum, and therefore the population of lions did not reach as low a minimum. From there, the sheep population slowly increased and the lion population slowly decreased until the two reach a stable point, with both populations being much lower than usual. The number of infected animals also remained stable.

Conclusion

Our first conclusion concerns the equilibrium conditions. We expected that the populations would achieve equilibrium with oscillations of smaller amplitudes. However, this proved wrong as the populations of both the sheep and lions varied greatly, albeit within a predictable pattern. This was consistent with our researched values, yet it still seems to defy common sense. However, this is what both the model and nature itself shows.

Next, the factory turned out to have a profound effect on the system. Both the sheep and lion populations were effected adversely by the factories, but the overall system achieved a new equilibrium. Indeed the new equilibrium had oscillations of smaller amplitude than the original equilibrium. In effect, the factory permanently damaged the system and reduced the net population, yet the ecosystem was able to partially recover and achieve a new equilibrium. This just demonstrates the resiliency of an ecosystem.

Next, we watched the effects hunters had on the ecosystem. Surprisingly, the hunters had the smallest, least noticeable effect. The lion population decreased significantly for a short while, yet the sheep population exploded. This is likely because the humans are less effective predators than the lions, hence when the lions died the sheep went essentially without predators for a while. After the hunters went away (the hunting season ended), the population returned to the exact same equilibrium that it previously had. From this, we conclude that as long as a population is not hunted to extinction, hunting will not have any severe long-term effects on an ecosystem.

The effects of drought are very profound. Both of the populations decrease and no longer achieve maximums as high as they previously did. However, as seen with the factory, the oscillations in population decrease in amplitude. Thus, as the resource availability decreases, the ecosystem can no longer sustain such wild fluxes in animal population. Rather, the animals achieve a static condition of almost no population change.

Finally, several conclusions can be drawn from the effects of the virus on the ecosystem. First, the populations achieved a new steady state after the virus was i9ntroduced, as with the drought and factory. This steady state consisted of oscillations of smaller amplitude. Perhaps more interesting, however, was the way the virus spread through the system. Even with only one infected animal initially, most of the ecosystem becomes infected after a short time. In addition, it takes a very long time for the virus to completely disappear from the system, as the populations continue to spread the virus with time. Perhaps in the future we can add in a method that allows us to take into account the fact that animals can build up immunity against the virus.

Achievements

The most significant original achievement of our project was the determination of the fact that as conditions limit resources, life span, or reproduction rates within an ecosystem, the amplitude of the population oscillations decreases significantly. This fact is demonstrated in the graphs above of the drought disaster, the factory disaster, and the virus disaster. We found it to be extremely interesting that the populations no longer oscillated wildly when forced to cope with harsh conditions. This is likely because under normal, ideal conditions, the ecosystem can safely sustain such fluctuations. However, when conditions become harsh, the populations can no longer fluctuate in an extreme way since the system can not sustain high populations.

References

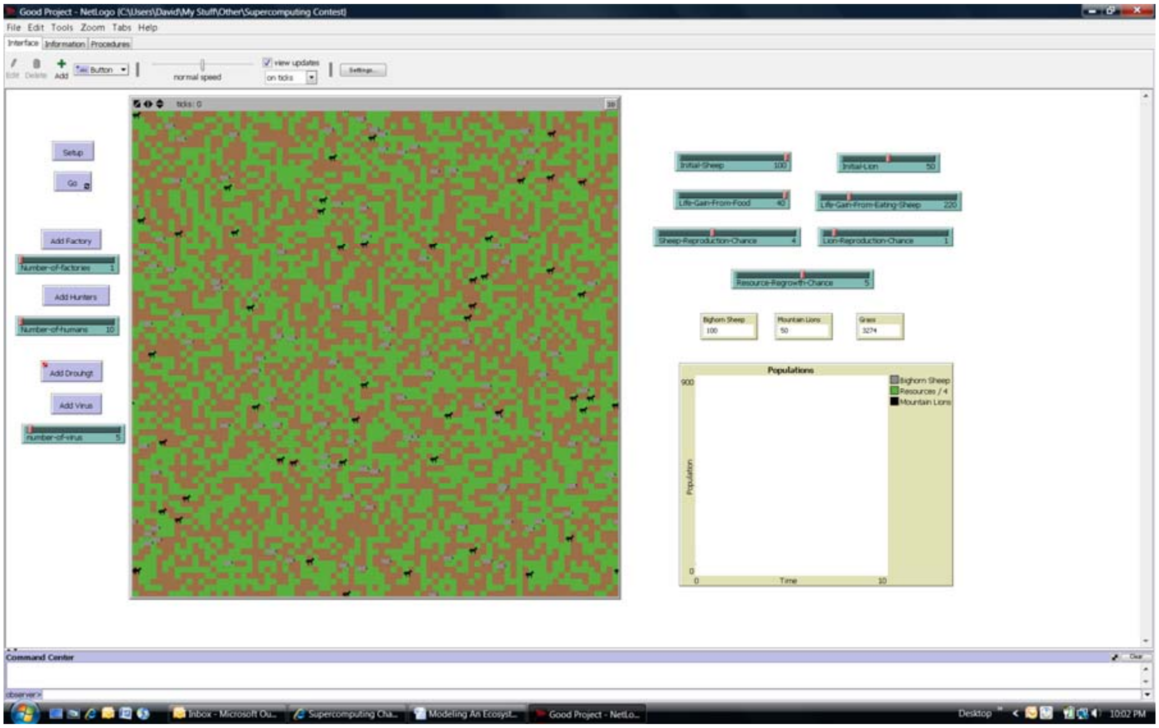
Books:

- Odum, Eugene P. Basic Ecology. Orlando: Saunders College Publishing, 1983.
- Ricklefs, Robert E. Ecology. New York: Chiron Press Incorporated, 1973.
- Nations, James D. Tropical Rainforests. New York: Franklin Watts Publishing, 1988.
- McCombs, Lawrence W., and Rosa Nicholas. What's Ecology?. Menlo Park: Addison-Wesley Publishing Company, 1973.

Websites:

- http://library.thinkquest.org/11353/ecosystems.htm
- http://www.1adventure.com/archives/cat_nature.html
- http://www.azgfd.gov/w_c/bhsheep/survey.shtml

Code




```
breed [bighornsheep sheep]

breed [mountainlion lion]

breed [human]

breed [factory]

breed [virus]

bighornsheep-own [life]

mountainlion-own [life]

human-own [life]

virus-own [life]


to setup

  clear-all
```

```
  ask patches [ set pcolor brown ]
  ask patches
   [
      set pcolor one-of [green brown]
     ]
  set-default-shape bighornsheep "sheep 2"
  create-bighornsheep Initial-Sheep
  [
   set color gray
   set size 2
   set life 50
   setxy random-xcor random-ycor
  ]
  set-default-shape mountainlion "wolf"
  create-mountainlion Initial-Lion
  [
   set color black
   set size 1.5
   set life 200
   setxy random-xcor random-ycor
  ]
  update-plot
end
```

```
to add-human
set-default-shape human "person"
create-human number-of-humans
[
set color red
set size 3
set life 1000
setxy random-xcor random-ycor
]
end

to add-factory
  set-default-shape factory "house"
  create-factory number-of-factories
  [
  set color red
  set size 8
  setxy random-xcor random-ycor
  ]
  decrease-reproduction
end
```

```
to add-drought
  if pcolor = green
  [
    set pcolor yellow
  ]
  set resource-regrowth-chance 2
end

to add-virus
  set-default-shape virus "virus-shape"
  create-virus number-of-virus
  [
    set color blue
    set size 5
    set life 1000
    setxy random-xcor random-ycor
  ]
end

to go
  if not any? turtles [ stop ]
  ask bighornsheep
  [
```

```
  move
  ifelse color = blue
  [
    set life life - 15
    infect
  ]
  [
    set life life - 10
  ]
  eat-food
  reproduce-sheep
  death
]
ask mountainlion
[

  move
  ifelse color = blue
  [
    set life life - 25
    infect
  ]
  [
```

```
   set life life - 10
  ]
  eat-sheep
  reproduce-lion
  death
]
ask human
[
  move
  set life life - 10
  kill
  death
]
ask factory
[
  kill-total
]
ask virus
[
  move
  infect
  set life life - 10
  death
```

```
  ]
  ask patches [ grow-grass ]
  tick
  update-plot
end

to move
  rt random 50
  lt random 50
  fd 1
end

to eat-food
  if pcolor = green or pcolor = yellow
  [
    set pcolor brown
    set life life + life-gain-from-food
  ]
end

to eat-sheep
  let prey one-of bighornsheep-here
  if prey != nobody
```

```
    [
      ask prey [ die ]
      set life life + life-gain-from-eating-sheep
    ]
end

to kill
  let prey one-of bighornsheep-here
  if prey != nobody
  [
    ask prey[die]
  ]
    let prey2 one-of mountainlion-here
  if prey2 != nobody
  [
    ask prey2[die]
  ]
end

to kill-total
  kill
end
```

```
to infect
 let prey one-of bighornsheep-here
 if prey != nobody
 [
  if 50 > random 101
  [
  ask prey[set color blue]
  ]
 ]
  let prey2 one-of mountainlion-here
 if prey2 != nobody
 [
  if 75 > random 101
  [
  ask prey2[set color blue]
  ]
 ]
end

to survival-chance
 if random 2 = 2
 [
  death
```

```
  ]
end
to decrease-reproduction
  set sheep-reproduction-chance sheep-reproduction-chance - .5
   set lion-reproduction-chance lion-reproduction-chance - .5
end

to reproduce-sheep
if sheep-reproduction-chance > 0
[
 if sheep-reproduction-chance >= random 101
 [
    set life (life / 2)                ;; divide energy between parent and offspring
    hatch 1 [ rt random-float 360 fd 1 ]   ;; hatch an offspring and move it forward 1 step
  ]
]
end

to reproduce-lion
if lion-reproduction-chance > 0
[
  if lion-reproduction-chance >= random 101
  [
```

```
    set life (life / 2)                ;; divide energy between parent and offspring
    hatch 1 [ rt random-float 360 fd 1 ]   ;; hatch an offspring and move it forward 1 step
  ]
]
end

to death
  if life < 0 [ die ]
end

to grow-grass
  if pcolor = brown
  [
    if resource-regrowth-chance >= random 101
      [
      ifelse any? patches with [pcolor = yellow]
      [
        set pcolor yellow
      ]
      [
        set pcolor green
      ]
      ]
```

```
  ]
end

to update-plot
  set-current-plot "Populations"
  set-current-plot-pen "Bighorn Sheep"
  plot count bighornsheep
    set-current-plot-pen "Mountain Lions"
  plot count mountainlion
  set-current-plot-pen "Resources / 4"
  ifelse any? patches with [pcolor = green]
  [
    plot count patches with [pcolor = green] / 4
  ]
  [
    plot count patches with [pcolor = yellow] / 4
  ]
end
```