The Water Robot

Team 35, Water Robot

New Mexico SuperComputing Challenge

Final Report

05 April 2023

Team Number: 35

School Name: Justice Code

Team Members: Aaliyha Maestas, Jenifer Batista Ortiz, Sara Santiago ,Alonda Jimenez Ramirez

Sponsor: Becky Campbell

Project Mentor: Dr. Anthony Lupinetti

Table of Contents

The Water Robot

Executive Summary

So we want to look at how to solve the problem of lots of trash being in the ocean. We think this is a problem becWe used the library of models in NetLogoWeb to choose a model which we could base

our project on so that we could make changes to code that is already started versus completely starting from scratch. Our net logo model started with a brown background, and ants that eat all of the brown stuff, and then when the ants eat the brown stuff then the background turns blue. It took a while - we want to hopefully turn the ants into circles. The color of the circles will be black. We already know how to change ause animals die from that trash and it causes more pollution in the water. Then we won't have good water and because the ocean food chain would be messed up and eventually it will affect all of us because it is bad for the environment. It could possibly lead to starvation, dehydration, and even eventually cannibalism. We want to use our computer model to show how this could happen. Then we want to be able to show how it could be prevented if less litter goes in the ocean. We want to make a waterproof robot that can clean underwater areas of litter and model the statistics that we figure out from a small real life test.

We have learned “very few life-sized humanoid robots are completely waterproof. Although there is an increasing interest in the study of aquatic robots, there are only a few successfully built aquatic robots. This is because of the challenging environment they work in and also because electricity and electronics essentially do not like water. Making a robot to work in water generally ends up in undesirable failures.Under water Crawling robots are marine versions of land based robots. These robots either roll on wheels or walk under water; if waterproofing is taken care, these robots can be easily designed and built. The concept of land based robots still holds good for underwater crawling, rolling or walking robots.”

We used the library of models in NetLogoWeb to choose a model which we could base our project on so that we could make changes to code that is already started versus completely starting from scratch. Our net logo model started with a brown background, and ants that eat all of the brown stuff, and then when the ants eat the brown stuff then the background turns blue. It took a while - wethe colors. We just need to know how to change the shape. Once we turn the shape into circles, we need to change the names. Then our model will be ready. want to hopefully turn the ants into circles. The color of the circles will be black. We already know how to change the colors. We just need to know how to change the shape. Once we turn the shape into circles, we need to change the names. Then our model will be ready.

What we hope to do with our model is to make sure that it can clean underwater and on land. We also hope that it can pick up 5 or 10% of trash in a lake that is kinda close to us so we don't have to worry about us walking far. We also want to make sure our model can go underwater without destroying itself. If the model destroys then it could not clean up the trash. We need to make our model water proof for it to go in the water without any problem.

Introduction

Problem Statement

We want to look at how to solve the problem of lots of trash being in the ocean. We think this is a problem because animals die from that trash and it causes more pollution in the water and then we won't have good water and because the ocean food chain would be messed up and eventually it will affect all of us because it is bad for the environment. It could possibly lead to starvation, dehydration, and even eventually cannibalism. We want to use our computer model to show how this could happen. Then we want to be able to show how it could be prevented if less litter goes in the ocean. We want to make a waterproof robot that can clean underwater areas of litter and model the statistics that we figure out from a small real life test.

We think a good solution would be making a water robot to go all the way down into the bottom of the ocean to pick up trash. It needs to withstand pressure with a good strong shape, like a circle. The expected result that we are looking for is our robot will be able to clean out at least 5% of the ocean trash and not be destroyed while doing it.

Background Research

We have been learning a lot about how to start coding using snappycode.org. We did research on how to make a robot move and how it can move under water. We are also learning how to code with Python and with Dr. Anthony. We have already learned how to make shapes and movements with code. We are beginning to use what we have learned to apply to our project so

we can write code to get a robot to move. The expected result that we are looking for is our robot will be able to clean out at least 5% of the ocean trash and not be destroyed while doing it.

Computational Model

Selection

We chose the original model from NetLogo called “Bug Hunt Consumers” and then we made changes so that the model would work to show our project. We want to show how a robot that could work underwater would be able to help clean up the waters of the world. First, we changed the color of the background to be blue so it was like water. Then, we changed the turtle color for the ants to black. Next, we need to code a way for the model to make the ants that are now our robots to move through the background which is now our water so that it could collect trash. We also need to add in turtles for the trash. We will need to look at effects from moisture and water currents, too.

Modifications

We started with the idea of cleaning up the ocean but we decided to start smaller with a body of water like a lake instead because of how long it would take a robot to travel the whole ocean. We used a method where we boxed out our tasks and made sure each group member had something to do. We found a model in netlogo web that we could use as a base and just made small changes like changing colors and the shapes of turtles and sliders so that we could work with the model. This was important because we are just learning to code but now we will be able to keep building on this and working on the project.

Visualization

We think our robot will be a circle with a mouth that can open and close when it eats trash. We are also hoping it can have a filter or it can spit out most of the water. We did research on how to make a robot move and how it can move under water. We are also learning how to code with Python and with Dr. Anthony. We have already learned how to make shapes and movements with code. We are beginning to use what we have learned to apply to our project so we can write code to get a robot to move.

Limitations

Our model can not (yet) have one robot. It has multiple robots and we are trying to figure out how to make it to where there's only one robot so it is realistic.Hopefully that happens soon which I am pretty sure it will.

Problem Solving Method

Verification & Corroboration

We tested our model multiple times to make sure it worked whenever we would make changes and it did. We made sure to run the model in between each change and go over step at a time to keep track of the code.

Conclusion

Results

When we added more trash the robots ate it but not as quickly and there was way more water. When we put less food the robots would eat they all basically sank to the bottom and didn't eat any trash. If we add less robots the trash still builds up and the robot still ends up sinking. If we add too many robots then all the trash is gone but the robots will end up eating each other because they will mistake each other for trash.

Discussion

We learned from our results that we need to change the model to make the “ants” into circles so they could look like robots.Then we will change the color to black. Once we do that then our model should be done and once it is done we can try to move the idea to a real robot.

Future Work

We want to change the number of robots that are on the code. Also the names in the lines of code need to match because to keep the model working.

Acknowledgments

We are really grateful for Ms. Campbell and Ms. Brown, our teachers, who helped us understand how to go through the process of doing a project for science fair. We are also thankful to our supercomputing reviewers, Max Lazo, Elizabeth Jimenez, Amy Knowles, and Patty Meyer. Our mentor,

Dr. Anthony helped us learn some coding and how to use NetLogo. We want to say thank you to our school district, Albuquerque Public Schools that gave us a grant to pay for our plane tickets.

References

http://www.robotplatform.com/knowledge/Classification_of_Robots/aquatic_robots.html

https://www.jpl.nasa.gov/news/robotic-navigation-tech-will-explore-the-deep-ocean

https://www.cnn.com/2022/07/30/tech/oceanone-diving-robot-scn/index.html

http://netlogoweb.org/launch#http://netlogoweb.org/assets/modelslib/Curricular%20Models/BEAGLE%20Evolution/Bug%20Hunt%20

https://www.whoi.edu/news-insights/content/underwater-robots-swarm-the-ocean/

https://www.americanscientist.org/article/the-robot-ocean-network

Appendix: Code

```
clear-all
 set bugs-born 0
 set bugs-died 0

 set bug-size 1.2
 set bugs-stride 0.3
 set bug-reproduce-age 20
 set min-reproduce-energy-bugs 10
 set max-bugs-offspring 2
 set max-bugs-age 100
 set grass-level 0

 set sprout-delay-time 25
 set grass-growth-rate 10
```

```
  set max-plant-energy 100


  set bugs-color (Black )
  set grass-color (Brown)
  set dirt-color (Blue)
  set-default-shape bugs "circle"
  set-default-shape embers "fire"
  add-starting-grass
  add-bugsreset-ticks
end

to add-starting-grass
  let number-patches-with-grass (floor (amount-of-grassland * (count patches) / 100))
  ask patches [
     set fertile? false
     set plant-energy 0
    ]
  ask n-of number-patches-with-grass patches  [
     set fertile? true
     set plant-energy max-plant-energy / 2
    ]
  ask patches [color-grass]
end

to add-bugs
  create-bugs initial-number-bugs  ;; create the bugs, then initialize their variables
```

```
 [
   set color bugs-color
   set size bug-size
   set energy 20 + random 20 - random 20 ;; randomize starting energies
   set current-age 0  + random max-bugs-age     ;; start out bugs at different ages
 set max-age max-bugs-age
   set #-offspring 0
   setxy random world-width random world-height
 ]
end



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; runtime procedures ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to go
 if ticks >= 1000 and constant-simulation-length? [stop]
  ask bugs [
   bugs-live
   reproduce-bugs
   ]
  ask patches [
   set countdown random sprout-delay-time
   grow-grass
   ]   ;; only the fertile patches can grow grass
  fade-embers
  age-disease-markers
```

```
  tick
end

to remove-a-%-of-bugs
  ;; procedure for removing a percentage of bugs (when button is clicked)
  let number-bugs count bugs
  ask n-of floor (number-bugs * bugs-to-remove / 100) bugs [
 hatch 1 [
     set current-age  0
     set breed disease-markers
     set size 1.5
     set color red
     set shape "x"
     ]
    set bugs-died (bugs-died + 1)
    die
  ]
end

to start-fire
  let current-grass-patches patches with [fertile?]
  let current-burn-patches n-of floor ( (count current-grass-patches) * grass-to-burn-down / 100)
current-grass-patches
  ask current-burn-patches [
    set countdown sprout-delay-time
    set plant-energy 0
```

```
    color-grass
    create-ember
    ]
end


to create-ember ;; patch procedure
  sprout 1 [
    set breed embers
    set current-age (round countdown / 4)
    set color [255 255 0 255]
    set size 1
    ]
end
to create-ember ;; patch procedure
  sprout 1 [
    set breed embers
    set current-age (round countdown / 4)
    set color [255 255 0 255]
    set size 1
    ]
end


to age-disease-markers
  ask disease-markers [
      set current-age (current-age  + 1)
      set size (1.5 - (1.5 * current-age  / 20))
```

```
      if current-age  > 25  or (ticks = 999 and constant-simulation-length?)  [die]
  ]
end


to fade-embers
  let ember-color []
  let transparency 0
  ask embers [
    set shape "fire"
    set current-age (current-age - 1)
    set transparency round floor current-age * 255 / sprout-delay-time
   ;; show transparency
    set ember-color lput transparency [255 155 0]
  ;;  show ember-color
    if current-age <= 0 [die]
    set color ember-color
  ]
end
to bugs-live
    move-bugs
    set energy (energy - 1)  ;; bugs lose energy as they move
    set current-age (current-age + 1)
    bugs-eat-grass
    death
end
```

```
to move-bugs
  rt random 50 - random 50
  fd bugs-stride
end


to bugs-eat-grass  ;; bugs procedure
  ;; if there is enough grass to eat at this patch, the bugs eat it
  ;; and then gain energy from it.
  if plant-energy > amount-of-food-bugs-eat  [
    ;; plants lose ten times as much energy as the bugs gains (trophic level assumption)
    set plant-energy (plant-energy - (amount-of-food-bugs-eat * 10))
    set energy energy + amount-of-food-bugs-eat  ;; bugs gain energy by eating


  ]
  ;; if plant-energy is negative, make it positive
  if plant-energy <=  amount-of-food-bugs-eat  [set countdown sprout-delay-time ]
end


to reproduce-bugs  ;; bugs procedure
  let number-new-offspring (random (max-bugs-offspring + 1)) ;; set number of potential offpsring from 1
to (max-bugs-offspring)
  if (energy > ((number-new-offspring + 1) * min-reproduce-energy-bugs)  and current-age >
bug-reproduce-age)
  [
     set energy (energy - (number-new-offspring  * min-reproduce-energy-bugs))      ;;lose energy when
reproducing --- given to children
```

```
      set #-offspring #-offspring + number-new-offspring
      set bugs-born bugs-born + number-new-offspring
      hatch number-new-offspring
      [
        set size bug-size
        set color bugs-color
        set energy min-reproduce-energy-bugs ;; split remaining half of energy amongst litter
        set current-age 0
        set #-offspring 0
        rt random 360 fd bugs-stride
      ]    ;; hatch an offspring set it heading off in a a random direction and move it forward a step
  ]
end

to death
  ;; die when energy dips below zero (starvation), or get too old
  if (current-age > max-age) or (energy < 0)
  [ set bugs-died (bugs-died + 1)
    die ]
end

to grow-grass  ;; patch procedure
  set countdown (countdown - 1)
  ;; fertile patches gain 1 energy unit per turn, up to a maximum max-plant-energy threshold
  if fertile? and countdown <= 0
     [set plant-energy (plant-energy + grass-growth-rate)
```

```
      if plant-energy > max-plant-energy
      [set plant-energy max-plant-energy]
      ]
  if not fertile?
    [set plant-energy 0]
  if plant-energy < 0 [set plant-energy 0 set countdown sprout-delay-time]
  color-grass
end

to color-grass
  ifelse fertile? [
    ifelse plant-energy > 0
    ;; scale color of patch from whitish green for low energy (less foliage) to green - high energy (lots of
foliage)
    [set pcolor (scale-color Brown plant-energy  (max-plant-energy * 2)  0)]
    [set pcolor dirt-color]
    ]
  [set pcolor dirt-color]
end



; Copyright 2011 Uri Wilensky.
; See Info tab for full copyright and license.
```