Space Junk: Problem of the Future

New Mexico

Supercomputing Challenge

Final Report

April 4, 2012

Team Members

Sofia Bali

Crystal Zamora

Gabriela Marchan

Susana Bali

Project Mentors

Rebecca Galves

Hadi Sharifi

Executive Summary

Our Netlogo model simulates a simplified version of space debris and collision that can occur in the Earth's geostationary orbit. It consists of turtles that represent functioning satellites, non-functioning satellites, and any man-made debris in orbit. The man- man debris turtles represent all fragments of launches that have accumulated over time, such as spent rocket stages, rocket boosters, and metals that are detached as the launches occur. In the model they orbit around the Earth and the collisions are counted. Our hypothesis was based off the Kessler Syndrome by NASA scientist Donald J. Kessler, which suggested that a critical density could be reached, in which colliding debris would cause a chain reaction of collisions that would threaten all unmanned space craft and make it very difficult to carry out further space exploration.

Problem Statement

The first man-made object to enter the Earth's orbit was the Soviet Union's Sputnik 1, in 1987. Since then, mankind's space exploration projects have filled the low-earth orbit and the geostationary earth-orbit with millions of objects and fragments (Stansbery, 2009).These objects, along with millions of naturally occurring micrometeorites, are a potential threat to future space projects and to the functional satellites in orbit. According to NASA (2010), they reach hypervelocity speeds of 22,000 mph; therefore even an object with a diameter of 1 to 10 cm can cause catastrophic damage to spacecraft due to the force of impact.

Our goal for this project is to create a simulation through NetLogo that will allow us to predict possible outcomes of the continuing growth of space junk. Through this model we will be able to determine whether or not certain variables can have positive results in controlling this growth. For example, we will determine the possible scenarios that can occur in the earth's orbit after a certain number of years. Fragments in orbit eventually are pulled into the atmosphere due to the gravitational pull of Earth or fall out of Earth's orbit because of its distance from Earth (Fleur, 2011). Will enough objects fall into the atmosphere to prevent chain reactions of collisions, or is the debris multiplying faster than it is being destroyed?

Method to Solve

Using NetLogo, our simulation attempts to show if a chain reaction of collisions might occur in space with the growing amount of debris. It also seeks to help the user see at what rates the debris is increasing. We began by using a simulation from the models library, "Circular Path example", which allowed us to see the code used to get the turtles to orbit around a center. The next step was to create different sized turtles to represent the different sizes of satellites and debris, excluding all the debris under 10 centimeters. When Setup begins, 165 debris are created. We found that there was an average of 16,500 debris above 10 centimeters orbiting in space. In the programming, we used the ratio of 100 to 1 in order to better portray this large quantity of objects. Setup also creates the non-functional and functional satellites. There are 74 non-functional satellites and 6 functional. There are 8,000 satellites in space, 7440 non-functional and 560 functional.

The next step was to get the debris to orbit. As mentioned previously, we used the circular path example from the models library to understand how the turtles could be commanded to move around a center point at random speeds. The speeds were set at random because each object has a different orbit and speed depending on its size and weight.

We wanted to take into consideration the gravitational pull of the Earth on the debris. Debris maintains an orbit that slowly gets closer to the Earth. Once it reaches the atmosphere, the particles burn up due to the high speeds, this is why most re-entries are destroyed before they can reach Earth's surface. How we demonstrated this in our model was by decreasing the radius of the turtles orbit by .05 for the debris 10 centimeters and larger, and .06 for the non-functional satellites, this difference due to weight and size determining how long it will take an object to orbit and reach Earth. This occurred every 1000 ticks, which represents a year.

The next variable to consider was the amount of new man-made objects being launched into space. Using the same ration of 1 to 100, we added 2.1 satellites every year in the simulation, because it is calculated that 210 new satellites will be launched per year for the next 10 years. Also added were 3 pieces of debris every 1000 ticks, which represent the 300 pieces of man-made debris that enters orbit annually.

A monitor, *collision sites*, was added to count the amount of collision. Once we knew the amount of collisions, we could determine when to add more debris. Every collision, 2 debris are added, representing 200 pieces of debris. This is based on the information that a piece of debris larger than 10 centimeters, when colliding with a satellite, would break into approximately 200 fragments. Again, we did not include any fragments measuring less than 10 centimeters, since those numbers reach millions.

Three other monitors were added to keep track of the amount of debris and satellites at a given moment: *Debris 10 centimeters and larger, non-functional satellites, and functional satellites.*

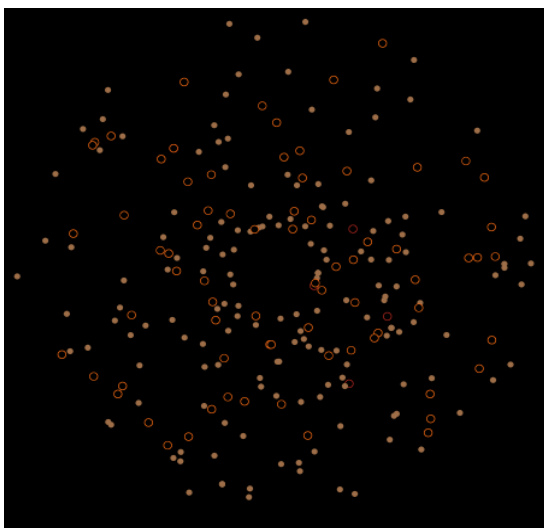
In order to show the collisions, we used the command *count turtles-here,* which checked if two turtles were in the same spot. If they are, the shape of the turtle and the color of the patches change to show the viewer where a collision has occurred. The turtle then dies, which represents an object that has broken into more pieces due to the collision.

Collisions begin once *collide* is pressed after Setup. This action had to be done separately using this button, since the commands for *collide* were turtle only, and the commands for *go* were observer only.

Most Significant Achievement

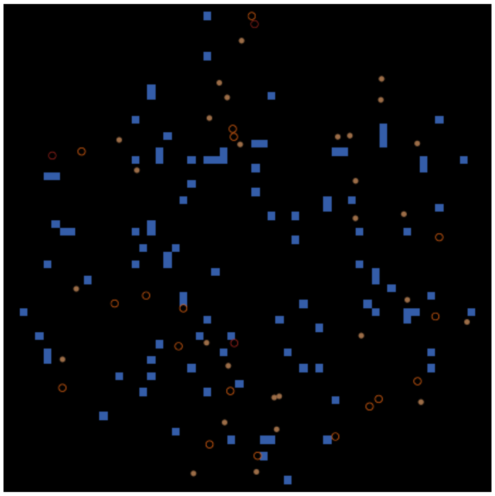
The hardest part was to get the debris to collide. Once we found that a separate button could be used to carry out this command parallel to the orbiting of the debris, we were able to start counting the collision of each type of turtle.

Results



Initial debris in orbit.

In the first 100 ticks, the most collisions occur. This is because the debris is more crowded since it was all added at the setup. After debris begin to disappear, they are added at a slower rate, therefore collisions occur with less frequency but more realistically.



Debris after 2,000 ticks (2 years)

The blue squares above represent the areas were a collision has occurred. Our hypothesis that collisions occur more frequently as time passes is supported by the program, because although they number fluctuates it begins to increase with more ticks.

Conclusion

Our program led us to conclude that debris is being added faster than it is being destroyed. Even though the program makes the turtles disappear once they collide, there are still more turtles in orbit than turtles dying. After 5,000 ticks (5 years), significant crowding of collision sites begins to be visible near the Earth's atmosphere. In the future it will be a problem to launch successful space missions, because there are too many hazards threatening the spacecrafts used. NASA has developed many materials and designs that are meant to protect the spacecraft. However, the hyper velocities and the large amount of debris makes it difficult to constantly steer the spacecraft out of harm's way.

References

National Aeronautics and Space Administration. (2010, April 26). Micrometeorites and orbital debris. Retrieved from http://www.nasa.gov/centers/wstf/laboratories/hypervelocity/mmod.html on November 30, 2011.

NASA Orbital Debris Program Office. (2009, July). Orbital debris frequently asked questions. Retrieved November 11, 2011 from http://earthobservatory.nasa.gov/IOTD/view.php?id=40173

Stansbery, E. (2009, July 22). Orbital debris surface examinations. Retrieved from http://orbitaldebris.jsc.nasa.gov/measure/surfaceexam.html on December 3,2011.

Space.com (2001, September 27). Complete coverage of NASA's fallen satellite UARS. Retrieved from http://www.space.com/13018-falling-nasa-satellite-uars-complete-coverage.html

Fleur, N.S. (2011, November 6). Debris may threaten future space endeavors. The Cornell Daily Sun. Retrieved from http://cornellsun.com/node/48932

Acknowledgements

We would like to thank our mentors Rebecca Galves and Hadi Sharifi, Dennis Trujillo, and the Young Women in Computing Staff.

Appendix- programming code

```
turtles-own [radius speed]
breed [ deadsatelites deadsatelite ]
breed [ debris10+ debri10+ ]
breed [ workingsatelites workingsatelite ]
breed [ spacestation ]
to setup
  clear-all

   repeat 74.4[
    ;; make  turtles, equally spaced in heading
    create-workingsatelites 1 [
      set radius random (max-pxcor - 5) + 5
      set speed random-float .1
      ;; move to edge of circle
      fd radius
      ;; turn to face clockwise
      rt 90
      ;; bigger turtles are easier to see
       set size 1
     set shape "circle 2"
     set color 25
    ]
  ]
```

```
repeat 165[
  ;; make  turtles, equally spaced in heading
  create-debris10+ 1 [
    set radius random (max-pxcor - 5) + 5
    set speed random-float .1
    ;; move to edge of circle
    fd radius
    ;; turn to face clockwise
    rt 90
    ;; bigger turtles are easier to see
    set size .7
    set shape "circle"
    set color 35
  ]
]

repeat 5.6[
  ;; make 12 turtles, equally spaced in heading
  create-deadsatelites 1 [
    set radius random (max-pxcor - 5) + 5
    set speed random-float .01
    ;; move to edge of circle
    fd radius
    ;; turn to face clockwise
    rt 90
    ;; bigger turtles are easier to see
     set size 1
   set shape "circle 2"
   set color 14
  ]
]
```

```
end

to go
  ask turtles [ debrie-orbit ]
  show count turtles
  tick
   moretrash

end

;; This is the core procedure.  It moves the turtle to the next point
;; on the circle, the given distance along the curve.

to debrie-orbit []  ;; turtle procedure
  if ( (ticks mod 1000) = 0 ) and (radius > 10) and breed = debris10+ [set radius (radius - .05)]
  ;; calculate how much of an angle we'll be turning through
  ;; (essentially converting radians to degrees)
  let theta speed * 180 / (pi * radius)
  ;; turn to face the next point we're going to
  rt theta / 2
  ;; go there
  fd speed
  ;; turn to face tangent to the circle
  rt theta / 2
  if (radius < 10) [ (die)  ]
  if ( (ticks mod 1000) = 0 ) and (radius > 10) and breed = deadsatelites [set radius (radius - .06)]
  ;; calculate how much of an angle we'll be turning through
```

```
  ;; (essentially converting radians to degrees)


end

to moretrash
   if ( (ticks mod 1000) = 0 )

 [
  repeat 3.05[
    ;; make 12 turtles, equally spaced in heading
    create-debris10+ 1
    [ set radius random (max-pxcor - 5) + 5
      set speed random-float .1
      ;; move to edge of circle
      fd radius
      ;; turn to face clockwise
      rt 90
      ;; bigger turtles are easier to see
      set size .7
      ;; thicker line is easier to see
      set pen-size 1
      ;; leave a trail
     set shape "circle"
     set color 35
    ]
  ]

  repeat 1.2[
    ;; make 12 turtles, equally spaced in heading
    create-workingsatelites 1
```

```
    [ set radius random (max-pxcor - 5) + 5
      set speed random-float .01
      ;; move to edge of circle
      fd radius
      ;; turn to face clockwise
      rt 90
      ;; bigger turtles are easier to see
       set size 1
      set shape "circle 2"
     set color 14
    ]
  ]
  ]
end
to collide
    if count turtles-here > 1
   [
    set shape "explosion"
    set size 5
    set pcolor blue
   ]
  if shape = "explosion"
  [
   wait .3
   die
  ]
   collision
end

to collision
  if count patches with [  pcolor = blue ] > 9 and count patches with [ pcolor = blue ] < 11
```

```
  [
   hatch-debris10+ 10

  ]

end
```