Modeling Ion Propulsion for Direct Space Craft Propulsion

New Mexico
Supercomputing Challenge
Final Report
April 3, 2007

Team 91
Rio Rancho Mid-High

Team Members

Robby Maura
Jarett Jones

Teacher

Debra Loftin

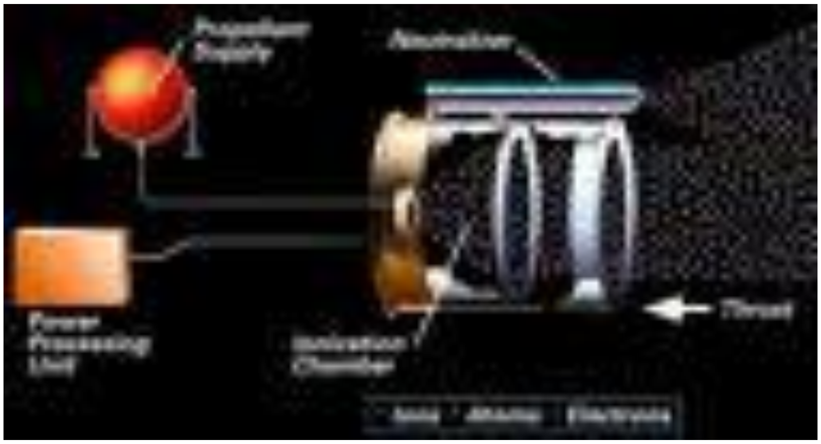
Project Mentor

Nick Bennet

Table of Contents

Introduction

Our Star Logo Program is designed to model the creation of ions for an ion propulsion engine that will be used in a spacecraft as a primary propulsion method. The program will show the xenon atoms crashing into electrons causing the atom to become an ion, then show the movement of the ion out of the ion generator into another part of the propulsion system. The program will collect data on the creation of ions how many are generated per minute and the efficiency of the system. If the model works as planned, we hope to see an ion production high enough to propel a spacecraft quickly and effectively.

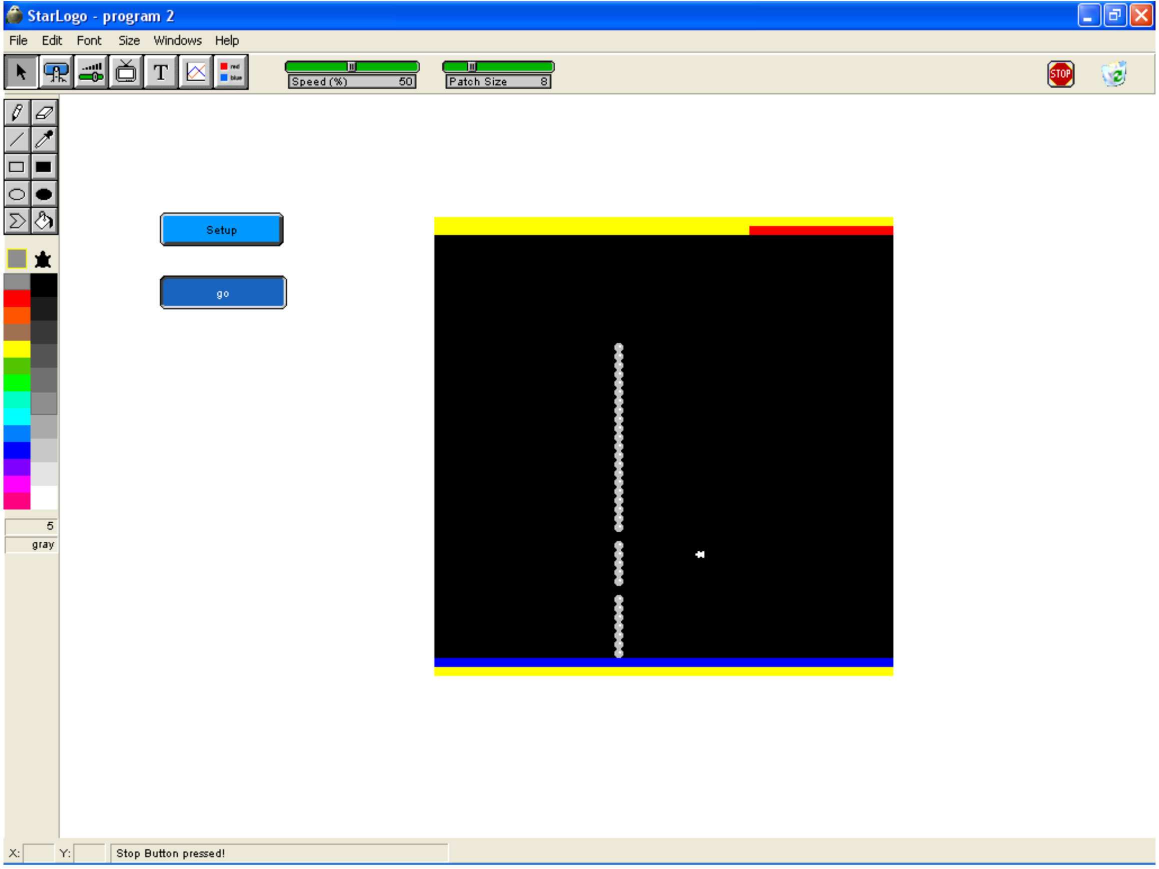


Executive Summary

Our program models the creation of ions to run an ion propulsion engine for a space craft. It shall demonstrate the process of making the ions as if on the actual space craft as it is flying threw space. This is called a cathode grid system.

Results

We do not have a complete program at this time, so therefore we have no results. Our expected results are that the ions that leave the chamber are so great in number and so fast that the spacecraft gets propelled forward. Here is a picture of what we have so far:



Speed (%) 50

Patch Size 8

5

gray

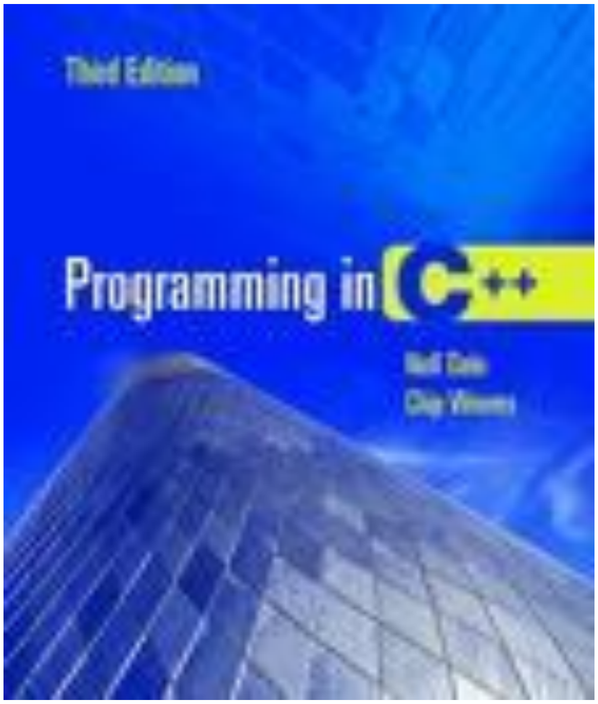
X: Y: Stop Button pressed!

Conclusion

Our program looks as if it will correctly model the ion generation but since it is not complete yet we can not tell any definite or finished results.

Recommendations

If we were to extend this project, we would focus more on the speed obtained by the ship than the propulsion, if the propulsion unit works. We would probably use a better program to program the ship, such as C++.



Acknowledgments

Our Mentor, Nick Bennett
Our parents
Our teacher sponsor, Ms. Loftin

Bibliography

Wright, Mike." over 50 years in the making." Ion Propulsion.1 Science@NASA. 12 Feb. 2007 <http://science.nasa.gov/newhome/headlines/prop06apr99_2.htm>. "Report." IPS Integrated Report.1 NASA. 12 Feb. 2007 <http://nmp-techval-reports.jpl.nasa.gov/DS1/IPS_Integrated_Report.pdf >.

Appendix

In the Turtle Procedures window:

```
to initialize-charge
        :x :y :magnitude
        setbreed charges
        setshape bubble-shape-color
        setxy :x :y
        setmagnitude :magnitude
        ifelse (magnitude < 0) [
                scale-color blue magnitude -120 20
        ] [
                scale-color red magnitude 120 -20
        ]
end


to random-unit
        output (random maxint) / maxint
end
to check
if xcor = abs (screen-half-width) [stop]
end
```

In the Observer procedures window:

```
breeds [electrons charges atom]
turtles-own [mass speed magnitude]
globals [ maxint step-number]
patches-own [fstrength fdirection]

to setup
        ca
        set step-number 0
        create-turtles-and-do 1 [setc yellow
        jump 25 rt 90 pd fd 50 bk 50 lt 90
        bk 1 rt 90 fd 50 bk 50 pu
        setc red fd 10 pd fd 15 pu bk 25 lt 90
        bk 19 bk 29 lt 90
        setc blue
        pd fd 50 bk 50 pu
        setc yellow
        rt 90 bk 1 lt 90 pd fd 50 die]
        set maxint (2 ^ 31 - 1)
        ask-patches [sprout [setbreed charges]]
                ask-charges [if pc = black [die] if pc = yellow [ set
        magnitude 0] if pc = red [ set magnitude 50] if pc = blue [ set
        magnitude -50]]
        ask-patches [sprout [setbreed atom]]
end
to compute-force
        ask-atom [if (count-charges-here = 1) [
                let [:x xcor]
                let [:y ycor]
                let [:sum-x 0]
                let [:sum-y 0]
                dolist [:charge list-of-charges] [
                        let [:magnitude (magnitude-of :charge)]

        let [:direction (towards-nowrap (xcor-of :charge) (ycor-of
        :charge))]
                        let [:distance (distance-nowrap (xcor-of :charge)
(ycor-of :charge))]
                        let [:sum-x (:sum-x + :magnitude * (sin :direction)
/ (:distance * :distance))]
                        let [:sum-y (:sum-y + :magnitude * (cos
:direction) / (:distance * :distance))]
                ]
                setfstrength (sqrt (:sum-x * :sum-x + :sum-y * :sum-y))
                setfdirection (180 + atan :sum-x :sum-y)]]
end
```

```
to go
        ct
        create-electrons-and-do 90 [
                setc blue
                setshape bubble-shape
                set mass 1 / 2000
                set speed 3
                seth 90
                setxy (0 - screen-half-width) (35 * random-unit - 23.5)
        ]
                crt 1
                ask-turtles [setxy screen-half-width (35 * random-unit - 23.5) setc white set mass 131 set speed 1 ]
                ask-turtles [seth 270]
                ask-electrons [seth 90]
                ask-turtles [repeat 51 [fd speed]check]

end
```