Deflect Lightning

Team 36, Deflect Lightning

New Mexico SuperComputing Challenge

Final Report

05 April 2023

Team Number: 36

School Name: Justice Code

Team Members:Alfonso Salas,Estevan Manriquez, Daniel Mason, Leland Martinez

Sponsor: Becky Campbell

Project Mentor: Dr. Anthony Lupinetti

Table of Contents

Deflecting Lightning

Executive Summary

The purpose of this project is to try to deflect lightning into generators .When there is a thunderstorm or a black out you can have electricity so you don't stay in the dark.Although you could use a candle just having light/electricity is way better to have.If I was in one of these situations me personally i don't have a lot of candles and i would assume that you would need a candle in each corner of each room or in the hallway especially in the hallway because since there is barley any source of light in the hallway and even if you “had light in the hallway” you won't because of the blackout or a lightning storm.

Some of the data and research that we have gotten is how to sort of deflect lightning but we haven't found out how to transfer the energy to a generator. The way we figured out to deflect the lightning is by using a lightning rod that is the way we found to deflect we also found how many volts you need to power a house but we don't know for how long that amount of volts is going to power the house for it could be minutes or even seconds and we also figured out how to change some of the free code in netlogo.web we also found out that New Mexico is very unlikely to have lightning storms. Our procedure for this project is to first find how we are going to transform the energy into the generator, how to get the power going, and how to keep the energy and electricity in there. We probably need a special type of generator. We are not sure for how long we are able to power the house for, minutes or even seconds and turn off right away.

In conclusion, we haven't actually made all the changes that we wanted to in our model next year we are hoping to be able to figure out how to make the model to do exactly what we want it to test so that we can see if harnessing the power of lightning in this way would actually work to help solve the energy crisis we are facing.

Introduction

Problem Statement:

Our project problem is researching ways to deflect lightning bolts .Whenever it rains and thunder strikes so it doesn't do damage, instead we can deflect it. This could change the amount of damage lighting does all around the world. The way that we can try to solve this problem is by making a lightning rod that connects to a generator and it can transfer energy.

Background Research

Some of the research and information that we have collected is mostly how to get lightning in to one place and some thing we have found is that copper can attract lightning. (https://www.copper.org/ 3/9/23) We have also done a lot of research on how lightning actually works and what causes lightning to strike and what we figured out that the negative charge at the bottom of the wants to link up with th positive charge. (https://scied.ucar.edu/ 3/9/23)

Computational Model

Selection

We chose to use a model from as a base for our code and then make changes to it. We have changed the colors of certain aspects of the display and the appearance of certain sprites. We changed the max number of a certain factor like dust or bigger clouds.

Modifications & Visualization

We have changed the colors of certain aspects of the display and the appearance of certain sprites. We changed the max number of a certain factor like dust or bigger clouds. We haven't made all the changes we wanted to yet. Some of the things we want to change are some of the sprites. We also want to add some sprites to the code.

Limitations

The model can't really deflect and conduct the lightning with a generator; the model also does not exactly show where the houses are. The model doesn't really have a exact place to go for example the lightning keeps going down until it hits the dust so if we can find away were we can make the lightning go through the dust and hit a house and somehow make the house glow to know that it got energy and also deflect some lightning from the generator so the generator does not explode.

Problem Solving Method

Verification & Corroboration

The model shows that in real life when lightning hits dust it has a way bigger flash and when it hits none of the dust particles it is just a blue thunder bolt it also shows that lighting usually goes for high objects like trees and houses. Whenever lightning hits water it would conduct it. A copper rod can stop and transport energy from a lightning bolt so you could store the energy in something to save it for when you need it. Iron, aluminum, gold, and silver can do the same. (https://www.ferrovial.com)

We want our model to demonstrate what happens when lightning hits a lightning rod. We are trying to simulate the collection of electricity with the model because it is too dangerous to do in person. If we can harness electricity this way it would be a good resource from lightning.

Conclusion

In conclusion, we haven't actually made all the changes that we wanted to in our model next year we are hoping to be able to figure out how to make the model to do exactly what we want it to test so that we can see if harnessing the power of lightning in this way would actually work to help solve the energy crisis we are facing.

Results:

The result that we are looking for is to be able to use a lightning rod to save energy in something like a generator. We also want to be able to deflect lightning with a lightning rod but the lightning rod won't be able to deflect by itself, so we would have to attach something so it would deflect.

Discussion:

We came up with this topic and how this benefits other people because we want to solve the energy crisis. We think this is a big problem because the world is quickly running out of fossil fuels and we need new energy sources. So far, we think we have done a good job with our model. We have learned a lot about how to code a computational model because we didn’t know how to code at all when we started. It was important to learn though because we can’t actually just stand outside and let ourselves get struck by lightning. We haven’t been able to run experiments with the model yet but this is going to be our next step as we keep working on our project.

Future Work

We would like to change a good amount of stuff of are model like changing the trees in to houses but that really it we still need to think of some changes that would represent on what we are trying to build.

Acknowledgments

We would like to acknowledge all the people that helped us to complete this project and get here. First, we want to thank our supercomputing reviewers, Max Lazo, Elizabeth Jimenez, Amy Knowles, and Patty Meyer. We also want to say thank you to our teachers, Ms. Campbell and Ms. Brown for hosting our after school club. We want to thank Dr. Anthony for teaching us code. Lastly, we want to say thank you and we love you to our families for all of their support and encouragement!

References

Atrash, Z. and Wilensky, U. (2011). NetLogo Lightning model. http://ccl.northwestern.edu/netlogo/models/Lightning. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

(https://www.ferrovial.com accessed 3-8-23)

(A cooper alliance member copper development association INChttps://www.copper.org/ 3-8-23)

(UCAR Center for science education https://scied.ucar.edu/3-9-23)

(UCAR Center for science education https://scied.ucar.edu/ 3-9-23)

Appendix: Code

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Variables ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

globals [
  cloud-line      ;; y coordinate of bottom row of cloud
  surface         ;; y coordinate of top row of earth
  top-of-cloud
  bottom-of-cloud
  dust-probability
  fade-probability
  lightning-struck?
  hit-tree?
]

breed [trees my-tree]
breed [clouds cloud]
breed [positive-cloud-ions positive-cloud-ion]
breed [negative-cloud-ions negative-cloud-ion]
breed [positive-ground-ions positive-ground-ion]
breed [step-leaders step-leader]
breed [positive-streamers positive-streamer]


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Setup Procedures ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; ;;;;;; Setup ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup
  clear-all
  setup-world
  reset-ticks
end

;; ;;;;;; Setup World ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; set up the world background
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-world

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;; set global variable values
  set cloud-line 100
  set surface -50; set the height of the earth
  set top-of-cloud 86
  set bottom-of-cloud (75 - (size-of-cloud))
  set dust-probability 70 ;; the probability the ionzation will happen to a dust particle
  set fade-probability 1 ;; the probability a path of ionized air will end or fade
  set lightning-struck? false
  set hit-tree? false

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;; set colors for the patches in the background
  ask patches [
```

```
  if pycor = surface ;; set color of the earth surface with input number of positive streamers
    [set pcolor green - 1]
  if pycor > (cloud-line - 1)
    [set pcolor gray]    ;;set color of the cloud
  if pycor > surface and pycor < cloud-line
    [set pcolor black]  ;; set color of the sky
  if pycor < surface
    [set pcolor green] ;; set color of the earth
 ]

 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; create the trees in the meadow based on the slider value for the number of trees
 create-trees number-of-trees [
  ;; half the trees are trees and half are pines
  ifelse ((random 2) = 0)
  [set shape "myTree"]
  [set shape "myPineTree"]
  set size ((random 20) + 15)
  setxy random-pxcor (surface + size / 2)
 ]

 ;;create cloud
 create-clouds 1 [
  set shape "cloud"
  set size (size-of-cloud * 20)
  setxy 0 75
 ]

 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;create Ions in the world and cloud
 create-positive-cloud-ions strength-of-field
 create-negative-cloud-ions strength-of-field
 create-positive-ground-ions strength-of-field

 ask positive-cloud-ions [
  set shape "+"
  set size 7
  setxy random-x-in-cloud (top-of-cloud - random-y-in-cloud)
 ]

 ask negative-cloud-ions [
  set shape "-"
  set size 7
  setxy random-x-in-cloud (bottom-of-cloud - random-y-in-cloud)
 ]

 ask positive-ground-ions [
  set shape "+"
  set size 7
  setxy random-x-in-cloud (surface - random 15) ;; distribute randomly along the surface, beneath the cloud area
 ]

 setup-streamers-leaders
 add-dust
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;create and place the streamers and leaders
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup-streamers-leaders

  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;create lightning makers
  create-step-leaders strength-of-field
  ask step-leaders [
    set shape "negativeStrm"
    set size 5
    let xcoor ((random (size-of-cloud * 15)) - (size-of-cloud * 7))
    setxy xcoor (bottom-of-cloud - (size-of-cloud * 2))
    facexy xcoor surface
  ]

  create-positive-streamers strength-of-field
  ask positive-streamers [
    set shape "positiveStrm"
    set size 5
    let xcoor random-x-in-cloud
    setxy xcoor surface
    while [(pcolor = lime) or (pcolor = turquoise) or (pcolor = brown)] [
      set ycor (ycor + 1)
    ]
    facexy xcoor bottom-of-cloud
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; The ions move around as a result of bouncing around and colliding (not explicitly modelled).
;; There is a charge separation within the cloud, the cause of which is not entirely understood,
;; although there are theories.  The separation is illustrated in this model, but the underlying
;; cause/process of the separation is not explicitly modelled.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to make-ions-move
  ask positive-cloud-ions [ ;; set the position to a random X, and change the Y by a random amount to stay in upper
area of cloud
    setxy random-x-in-cloud (top-of-cloud - random-y-in-cloud)
  ]
  ask negative-cloud-ions [;; set the position to a random X, and change the Y by a random amount to stay in lower
area of cloud
    setxy random-x-in-cloud (bottom-of-cloud - random-y-in-cloud)
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; determine the area of the cloud
;;; and randomly select positions from width and height (X and Y)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; get a random X coordinate within the cloud's area
to-report random-x-in-cloud
  ;; the width is determined by multiplying the size-of-cloud variable by 15
  ;; however, we subtract a multiple of the size to make the distribution cover the positive and negative regions of the
axis
  report random (size-of-cloud * 15) - (size-of-cloud * 7)
end
```

```
;; get a random Y coordinate within the cloud's area
to-report random-y-in-cloud
  ;; the height of the cloud is determined by doubling the size-of-cloud variable
  report random (size-of-cloud * 2)
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; disperse dust into the air
;;; these particles influence the path of the lightning
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to add-dust
  ask n-of dust patches with [pycor > surface  and pycor < cloud-line] [
    set pcolor gray - 2
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Runtime Procedures ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go
  ;; if lightning has struck, clear the faded paths as the charge has now been neutralized
  if lightning-struck? [
    clear-fades ;; if there's been a lightning strike, clear all the faded paths
    stop
  ]

  ;; if the leaders and streamers have all died or faded, stop the model
  if (not any? step-leaders) and (not any? positive-streamers) [
    stop
  ]

  ;; since the positive streamers grow at a slower rate than the step leaders,
  ;; we keep track of the time so make them move at appropriate times compared to each other

  ;; the step leaders leave the cloud and move towards the earth at one patch per tick
  move-step-leaders-down

  ;; meanwhile, the positive streamers extend from the earth towards the cloud, but at a slower rate
  if ticks > 50
    [grow-positive-streamers-up]

  ;; as time moves on the ions move around the cloud
  make-ions-move

  tick
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; get the step leaders to move towards the earth
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to move-step-leaders-down
  let move-x 0
  let move-y 0

  ask step-leaders[
    set move-x 1000 ;; default values to indicate that the move coordinates have not been set
```

```
set move-y 1000

ask neighbors with [pcolor = gray - 2] [ ;; if the patch is gray then the leader
                                         ;; has reached a dust particle and is more likely
                                         ;; move that direction and ionize it
  if random 100 < dust-probability [
    set move-x pxcor
    set move-y pycor
  ]
]

;; if the stepleaders sense a positive streamer they move towards it
let xcoor xcor
let ycoor ycor

;; so we check for positive streamers between the leader and the ground
ask patches in-cone 10 60 [

  ifelse any? trees-here ;; if any tree exists in the local area
  [
    ifelse ((xcoor - pxcor) > 0)   ;; check if it is to the left
      [set move-x (xcoor - 1)] ;; then move one step to the left
      [set move-x (xcoor + 1)] ;; otherwise, it is to the right, so move one step to the right

    set move-y (ycoor - 1) ;; move down one step closer to the surface
  ]
  [
    if any? positive-streamers-here ;; if one exists then the leader moves in the direction towards it
      [ ifelse ((xcoor - pxcor) > 0)   ;; if the positive streamer is to the left
        [set move-x (xcoor - 1)] ;; then move one step to the left
        [set move-x (xcoor + 1)] ;; otherwise, it is to the right, so move one step to the right

      set move-y (ycoor - 1) ] ;; move down one step closer to the surface
  ]]

;; if no charges or dust particles are encouraging direction, pick at random but most likely toward earth
;; the random numbers are used to show the probabilities of ionized air moving in specific directions.
;; the path will move based on some probability of ten by selecting a random number below 10.
;; these probabilities are not scientific and are used only to help guide the path

if move-x = 1000 [

  ;;set y movement
  let rand-no random 10

  ;; if that number is greater than 2, the path will move directly down one towards earth (80%).
  ;; if it is less than 2 the path won't move (10%)
  ;; if it is 2 the path will move up one towards the cloud (10%)

  ifelse (rand-no > 2)[
    set move-y ycor - 1
  ][
  ifelse (rand-no < 2)[
    set move-y ycor
  ][
  set move-y ycor + 1
```

```
    ]
    ]

    ;; if that number is greater than 6, the path will move to the left (40%).
    ;; if it is less than 2 the path won't move (20%)
    ;; if it is between 2 and 5 the path will move to the right (40%)

    ;;set x movement
    set rand-no random 10

    ifelse (rand-no > 6)[
      set move-x xcor - 1
    ][
    ifelse (rand-no < 3)[
      set move-x xcor
    ][
    set move-x xcor + 1
    ]
    ]
  ]

  ;; make sure its still in the bounds of the world; otherwise, the path fades
  ifelse (move-x < min-pxcor) or (move-x > max-pxcor) or (move-y > max-pycor)
    [die]
    [if (random 100 < fade-probability) ;; these is some chance (the fade-probability) that the path will fade on its
own
      [die]
    setxy move-x move-y
    ]

  ;; check if the path hit a tree
  did-hit-tree

  ;; and change whatever patch the leader has moved to plasma, or a blue patch
  ask patch-here [
    ifelse (any? positive-streamers-here or  ;; if that patch is a positive streamer, or
      (pcolor = (violet + 3)) or             ;; the path of a streamer, or
      (hit-tree?) or                         ;; a tree, or
      (pycor = surface))                     ;; the earth's surface
    [ make-lightning move-x move-y           ;; then lightning strikes and we turn the patch yellow
      set pcolor yellow ]
    [ set pcolor (blue + 3)]                 ;; otherwise, turn that patch of air to plasma (ie, a blue patch)
  ]
 ]
end


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; here we approximate the position of trees using a circle roughly the size of the trees
;; if the step leader path reaches the tree, lightning strikes
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to did-hit-tree
 ask patches in-radius 12 [
  if any? trees-here
    [ set hit-tree? true ]
 ]
```

```
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; grow the positive streamers toward the cloud
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to grow-positive-streamers-up

  ask positive-streamers [
    let move-x 0
    let move-y 0

    ;;;;;;;;;;;;;;;;;; if the patch is gray then the leader has reached a dust particle and is more likely to ionize it
    ask neighbors [
      if pcolor = gray - 2 [
        if random 100 < dust-probability [
          set move-x pxcor
          set move-y pycor
        ]
      ]
    ]

    ;;;;;;;;;;;;;;;; generate random direction moves for the path to grow
    if (move-x = 0)[
      ;;set y movement
      let rand-no random 10

      ifelse (rand-no < 8)[
        set move-y ycor + 1
      ][
      ifelse (rand-no < 2)[
        set move-y ycor
      ][
      set move-y ycor - 1
      ]
      ]

      ;;set x movement
      set rand-no random 11
      ifelse (rand-no < 5)[
        set move-x xcor - 1
      ][
      ifelse (rand-no < 2)[
        set move-x xcor
      ][
      set move-x xcor + 1
      ]
      ]
    ]

    ;;;;;;;;;;;;;;;; if the movement is off the screen, kill the stream
    let random-growth-height (surface + random 20 + 20)

    ifelse (move-x < min-pxcor) or (move-x > max-pxcor) or
      (move-y < surface) or (move-y > max-pycor) or
      (move-y > random-growth-height)
      [die]
```

```
   [if (random 100 < fade-probability)  ;;there is some probability that the stream will just end
     [die]
   setxy move-x move-y
   ]

  ;;;;;;;;;;;;;;;;;; if the patch is empty then change the color to show the path is now positive
  ;;;;;;;;;;;;;;;;;; if the patch has both a stepleader and a positive streamer the paths connect and lightning strikes!
  ask patch-here [
   ifelse (any? step-leaders-here or (pcolor = (blue + 3)))
     [make-lightning move-x move-y
       set pcolor yellow ]
     [set pcolor (violet + 3)]
  ]

 ] ;; end ask +streamers
end

;;; ;;;;;;; Make Lightning ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; we'll check all the neighboring patches to see if they've been ionized (or turned light blue color)
;; if they have been they are now charged and send it to their neighbors as well
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to make-lightning [x y]
 set lightning-struck? true
 create-strike x y
end

to create-strike [x y]
 if (y < bottom-of-cloud) and (y > (surface - 1)) [
  ask neighbors [
   if (pcolor = (violet + 3)) or (pcolor = (blue + 3)) [
     set pcolor yellow
     make-lightning pxcor pycor
   ]
   if pcolor = black [
     set pcolor white
   ]
  ]
 ]
end


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;clear the remaining positive streamers from the sky
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to clear-fades
 ask patches [
  if (pcolor = (violet + 3)) or (pcolor = (blue + 3)) [
   setpcolor black
  ]
 ]
 ask positive-streamers [ die ]
 ask step-leaders [ die ]
end
```