ΑΓΓΕΛΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ ΑΜ 1067345

ΤΣΙΡΩΝΗΣ ΑΝΔΡΕΑΣ ΑΜ 1063428

Τρίτο σύνολο Θεωρητικών και Προγραμματιστικών ασκήσεων Αποκεντρωμένου υπολογισμού

Περιεχόμενα

[Θεωρητικές ασκήσεις 1](#_Toc170487065)

[Άσκηση 1 1](#_Toc170487066)

[Άσκηση 2 2](#_Toc170487067)

[Προγραμματιστικές Ασκήσεις 2](#_Toc170487068)

[Άσκηση 1 3](#_Toc170487069)

[Περίπτωση όλοι οι πράκτορες να μην έχουν καθόλου επιμονή στην αρχική γνώμη τους 5](#_Toc170487070)

[Περίπτωση όλοι οι πράκτορες να έχουν πλήρη επιμονή στην αρχική γνώμη τους 6](#_Toc170487071)

[Περίπτωση όπου μόνος ένας πράκτορας έχει επιμονή στην αρχική γνώμη και όλοι οι άλλοι δεν επιμένουν καθόλου στην αρχική γνώμη τους 7](#_Toc170487072)

[Τυχαία επιλογή της επιμονής του κάθε πράκτορα 9](#_Toc170487073)

[Τελικές παρατηρήσεις 10](#_Toc170487074)

[Άσκηση 2 10](#_Toc170487075)

[Υποερώτημα 1) 11](#_Toc170487076)

[Υποερώτημα 2) 11](#_Toc170487077)

[Υποερώτημα 3) 12](#_Toc170487078)

[Υποερώτημα 4) 12](#_Toc170487079)

[Υποερώτημα 5) 13](#_Toc170487080)

[Υποερώτημα 6) 14](#_Toc170487081)

[Υποερώτημα 7) 14](#_Toc170487082)

[Κώδικας 15](#_Toc170487083)

[Άσκηση 1 16](#_Toc170487084)

[Άσκηση 2 21](#_Toc170487085)

# Θεωρητικές ασκήσεις

## Άσκηση 1

1. To μητρώο Α που αναπαριστά το σύστημα, με είναι το εξής:

Για να είναι στοχαστικό το μητρώο προς τις γραμμές, πρέπει το άθροισμα των στοιχείων των γραμμών να είναι ίσο με ένα. Για την πρώτη γραμμή έχουμε

1. Γραμμή 1 = 1
2. Γραμμή 2 = α + 1 – α = 1

Οπότε το μητρώο είναι στοχαστικό προς τις γραμμές

1. Οι ιδιοτιμές και τα ιδιοδιανίσματα του Α θα βρεθούν με τους εξής παρακάτω τύπους:

Καταρχάς θα ψάξουμε για τις ιδιοτίμες, οπού θα τις βρούμε μέσω του τύπου:

Όπου θα διαμορφωθεί έως εξής:

Λύνουμε την διαφορική εξίσωση έως:

Οπού θα έχουμε τις ιδιοτιμές για το μητρώο Α:

Θα βρούμε τώρα τα ιδιοδιανύσματα λύνοντας για

Όπου το σύνολο εξισώσεων θα είναι έως εξής:

Το οποίο θα καταλήξει στο ιδιοδιάνυσμα .

Για ,θα έχουμε και πάλι το ίδιο ιδιοδιάνυσμα, το . Με αυτό υποθέτουμε ότι υπάρχει μόνο μία μοναδική ιδιοτιμή.

3)

a

X2

X1

1-a

1

Το γράφημα είναι ελαφριά συνεκτικό, καθώς ενώ υπάρχει διαδρομή από το x1 στο x2, δεν υπάρχει διαδρομή από το x2 στο x1, καθώς το x1 εξαρτάται μόνο από τον εαυτό του.

4)Για να βρούμε που συγκλίνει ο αλγόριθμος ως συνάρτηση των αρχικών παικτών, θα πρέπει να δούμε εάν μπορεί να συγκλίνει ο αλγόριθμος. Στα συστήματα μέσου όρου, για να έχουμε σύγκλιση πρέπει να ισχύει ισχυρή απεριοδικότητα, στοχαστικότατα και ισχυρή συνδεσιμότητα. Ενώ ισχύουν τα δύο πρώτα, είδαμε στο πάνω ερώτημα ότι δεν υπάρχει ισχυρή συνδεσιμότητα. Οπότε δεν θα υπάρχει σύγκλιση σε κάποια τελική τιμή

## 

## Άσκηση 2

a) Το μητρώο Ά που αναπαριστά το σύστημα, με ,είναι το εξής:

Με το συνολικό σύστημα να είναι το:

Ξέρουμε ότι το μητρώο είναι στοχαστικό προς τις γραμμές, γιατί σε κάθε γραμμή έχουμε μία ξεχωριστή τιμή αυτοπεποίθησης και συμπληρωματική της προς 1 . Ξέρουμε ότι οι τιμές αυτοπεποίθησης είναι ίσο και μεγαλύτερο του μηδενός, οπότε το αποτελέσματα της κάθε γραμμής θα είναι πάντα ένα.

b) Το μητρώο θα είναι έως εξής:

Αφού έχουμε την σχέση , μπορούμε να υποθέσουμε τις δύο παρακάτω σχέσεις που θα μας βοηθήσουν να απλοποιήσουμε το παραπάνω μητρώο :

Οπότε το μητρώο θα μετατραπεί έως εξής:

c) Για τα ιδιοτιμές, θα χρησιμοποιήσουμε τον τύπο που χρησιμοποιήσαμε και πριν:

Οπότε

Θα χρησιμοποιήσουμε την σχέση οπότε:

Οπότε και

Τα προς δεξιά ιδιοδιανύσματα, με λ να υπονοεί και την και την ,άρα θα είναι δύο ιδιοδιανύσματα θα είναι η θα είναι:

Όπου το σύνολο εξισώσεων, θα είναι έως εξής:

Μπορούμε να δούμε από την δεύτερη γραμμή, ότι μπορούμε να διαμορφώσουμε και τα δύο ιδιοδιανύσματα έως εξής:

Το οποίο θα μας δώσει τα δεξιά ιδιοδιανύσματα v , με λ ένα από τα δύο λ που βρέθηκαν.

Τα αριστερά ιδιοδιανύσματα u δίνονται από την σχέση:

Οπότε

Όπου το σύνολο εξισώσεων, θα είναι έως εξής:

Μπορούμε να δούμε από την δεύτερη γραμμή, ότι μπορούμε να διαμορφώσουμε και τα δύο ιδιοδιανύσματα έως εξής:

Το οποίο θα μας δώσει τα δεξιά ιδιοδιανύσματα u , με λ ένα από τα δύο λ που βρέθηκαν.

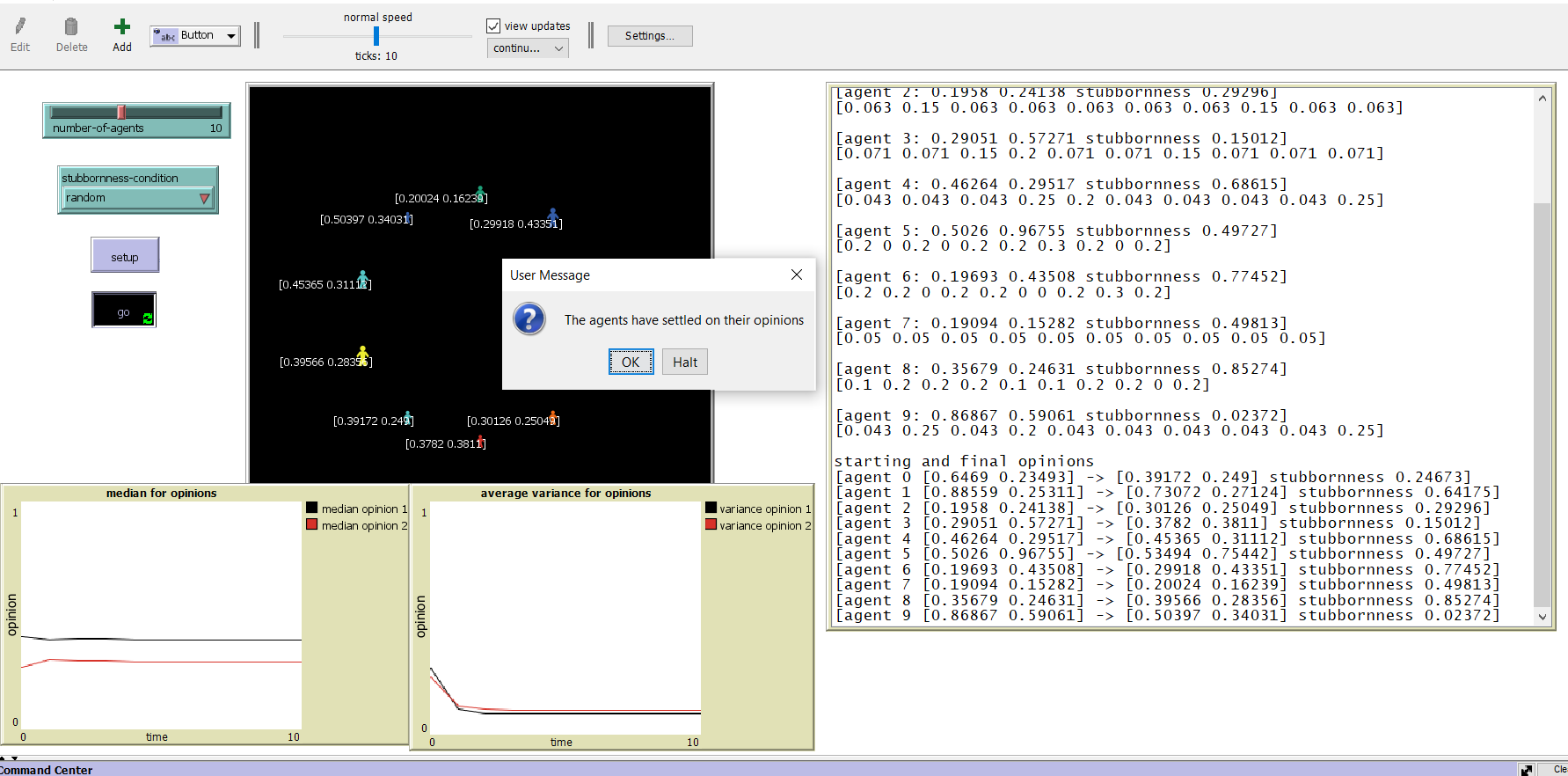
d) Η σύγκλιση του συστήματος θα τείνει στην τιμή ,που διαμορφώνεται από την παρακάτω σχέση, επειδή είναι θ-connected.

Με V να είναι:

Μπορούμε να δούμε ότι όσο πιο μεγάλη είναι η αυτοπεποίθηση ενός πράκτορα στην γνώμη του, τόσο μειώνεται η επιρροή που έχει πίνακας του συστήματος στις γνώμες των πρακτόρων, και αναλόγως, η επιρροή που ασκεί ή αρχική γνώμη του κάθε χρήστη αυξάνεται.

# Προγραμματιστικές Ασκήσεις

## Άσκηση 1



Υλοποιήσαμε το Friedkin-Johnsen Model όπως δόθηκε στην διαφάνεια 61 στο μάθημα των opinion Dynamics. Το μοντέλο είναι μία εξέλιξη του DeGroot μοντέλου, με την διαφορά ότι κάθε πράκτορας έχει μία επιμονή (stubbornness) στην αρχική του άποψη. Με βάση την υλοποίηση του transition matrix, θεωρούμε ότι μοιράζουμε την επίδραση που έχουν πράκτορες στην άποψη ενός συγκεκριμένου πράκτορα έως το πολύ έως και 3 ομάδες κάποιου αριθμού πρακτόρων, όπου ο καθένας πράκτορας από κάθε έχει μία συγκεκριμένη άποψη, αυτή της ομάδας που τέθηκε. Οι υπόλοιποι είναι μηδέν. Μπορεί να υπάρχουν λιγότερες από 3 ομάδες με μη μηδενικά στοιχεία. Οι 3 ομάδες θα πάρουν το 50%,30% και 20% της επιρροής σε ένα κόμβο.

Δεν φτιάξαμε ολοκληρωμένο transition matrix, αλλά επειδή οι γραμμές στο μητρώο αντιστοιχούν στο κάθε πράκτορα και οι στήλες τον συντελεστή επίδρασης των υπόλοιπων πρακτόρων, υλοποιήσαμε την κάθε γραμμή στον αντίστοιχο πράκτορα ξεχωριστά. Χρησιμοποιήσαμε το output window για εμφάνιση των τελικών αποτελεσμάτων και της αρχικής κατάστασης του μοντέλου και το transition matrix. Σε κάθε πράκτορα στο world view απεικονίζεται η τωρινή του τιμή. Επίσης, για να επιλέξουμε μία από τις επιλογές που εξετάζουμε, θέτουμε την ανάλογη τιμή στο stubbornness-condition.

Αναπαραστήσαμε το μέσο όρο και την διασπορά με τα αντίστοιχα primitive commands, με μόνη προσθήκη στο ότι πολλαπλασιάζουμε επί τέσσερα, για να κανονικοποιήσουμε την διασπορά και να μπορεί να φανεί σωστά σε μία μεταβλητή από τιμές μηδέν έως ένα. Το πολλαπλασιάζουμε επί 4 καθώς το 1/4 είναι η μέγιστη διασπορά για μηδέν έως ένα τιμή. Ουσιαστικά διαιρούμε την διασπορά με το 1/4,που είναι ίδιο με τον πολλαπλασιασμό επί 4

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, οθόνη

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, αριθμός

Περιγραφή που δημιουργήθηκε αυτόματα

Στις αναλύσεις που θα κάνουμε, χρησιμοποιούμε 10 πράκτορες, που δεν είναι πολλοί, αλλά είναι αρκετοί για να δημιουργήσουν μία αρκετά περίπλοκη σχέση μεταξύ τους, οπότε δεν θα μπούμε σε μεγάλη λεπτομέρεια να ψάξουμε να βρούμε πως κάθε πράκτορας φτάνει στην τελική του άποψη. Θα εστιάσουμε στο κομμάτι που αφορά κυρίως την επιμονή στην αρχική άποψη.

Επίσης το μέγεθος του πράκτορας στο world view βασίζεται από το μέγεθος της επιμονής του κάθε πράκτορα, για οπτική εμφάνιση.

### Περίπτωση όλοι οι πράκτορες να μην έχουν καθόλου επιμονή στην αρχική γνώμη τους

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα**

Μπορούμε να δούμε από πολλαπλές επαναλήψεις ότι αν δεν υπάρχει επιμονή στην αρχική γνώμη από τους πράκτορες, όλοι οι πράκτορες θα καταλήξουν σε μία ουσιαστικά κοινή απόφαση, οπού κατά μεγάλη πλειοψηφία είναι κοντά στις άκρες, είτε στο μηδέν είτε στο ένα. Μπορούμε να το καταλάβουμε από το γεγονός ότι η διασπορά είναι πολύ μικρή, σχεδόν ελάχιστη Όποτε μπορούμε να δούμε ότι πράγματι υπάρχει ομοφωνία μεταξύ των πρακτόρων. Όλοι μαζί θα οδηγηθούν είτε προς το μηδέν είτε προς το 1, το οποίο εξαρτάται από τις αρχικές τους γνώμες και τις σχέσεις μεταξύ τους.

Επίσης θα δούμε και έπειτα ότι για να καταλήξουν σε μία τελική άποψη όλοι οι πράκτορες, παίρνει πολύ παραπάνω χρόνο από ότι στο να υπάρχει έστω και ένα πράκτορας με κάποια επιμονή στην αρχική του γνώμη. Οι μόνες γνώμες που δεν είναι στις άκρες, μπορούμε να δούμε ότι πήραν πάρα πολύ χρόνο να φτάσουν στην τελική τους άποψη οι πράκτορες, αλλά δεν μπορούμε να σιγουρευτούμε ότι δεν είναι τυχαίο.

|  |  |
| --- | --- |
|  |  |
|  |  |

### Περίπτωση όλοι οι πράκτορες να έχουν πλήρη επιμονή στην αρχική γνώμη τους

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, εικονίδιο υπολογιστή

Περιγραφή που δημιουργήθηκε αυτόματα**

Μπορούμε να δούμε ότι αν όλοι οι χρήστες έχουν απόλυτη επιμονή στην αρχική τους άποψη, δεν θα υπάρξει κανενός είδος συνδιαλλαγής μεταξύ των πρακτόρων που να αλλάξει την άποψη τους, όποτε θα έχουν πάντα την ίδια γνώμη.

Μπορούμε να δούμε ότι οι μέση τιμή είναι διαμερισμένη σε τιμές από μηδέν έως ένα, οπού βασίζεται απόλυτα από τις τιμές που έδωσε η random-float στις γνώμες του πράκτορα, με λίγες μεσαίες τιμές να είναι στις ακραίες γνώμες όμως .Η διασπορά είναι αρκετά μεγαλύτερη.

|  |  |
| --- | --- |
|  |  |
|  |  |

### Περίπτωση όπου μόνος ένας πράκτορας έχει επιμονή στην αρχική γνώμη και όλοι οι άλλοι δεν επιμένουν καθόλου στην αρχική γνώμη τους

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, ιστοσελίδα

Περιγραφή που δημιουργήθηκε αυτόματα**

Μπορούμε να δούμε ότι οι πράκτορες τείνουν στην γνώμη του πράκτορα που μένει απολύτως σταθερός στην αρχική του, οπού προκύπτει και ομοφωνία πάνω στην γνώμη αυτή. Αυτό μπορούμε να το διαπιστώσουμε από την μέση τιμή και διασπορά διάφορων επαναλήψεων. Μπορούμε να το διαπιστώσουμε ότι συμβαίνει αυτό, από το γεγονός ότι έχουμε πολλές διαφορετικές τιμές της μέσης τιμής αλλά πολύ μικρή διασπορά στις γνώμες των πρακτόρων. Μπορούμε να υποθέσουμε βάσιμα ότι οι μέσες τιμές είναι η γνώμη του επίμονου πράκτορα.

Παρατηρούμε επίσης την πολύ μεγαλύτερη ταχύτητα σύγκλισης σε σχέση με την περίπτωση της απολύτου μη επιμονής πρακτόρων, η οποία όμως δεν είναι τόσο κοντά στην απόλυτη επιμονή πρακτόρων.

|  |  |
| --- | --- |
|  |  |
|  |  |

### Τυχαία επιλογή της επιμονής του κάθε πράκτορα

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, ιστοσελίδα

Περιγραφή που δημιουργήθηκε αυτόματα**

Εδώ βλέπουμε ότι έχουμε μία λιγότερο ακραία εκδοχή του να έχουν όλοι οι πράκτορες απόλυτη επιμονή. Μπορούμε να δούμε ότι οι μέσες τιμές είναι κάπου στις μεσαίες τιμές μεταξύ του 0 και 1, ακόμα ποιο κοντά στην μέση από την απόλυτη τυχαία επιλογής της απόλυτης επιμονής. Η διασπορά είναι μικρότερη από την απόλυτη επιμονή, αλλά και πάλι μεγαλύτερη από τις περιπτώσεις οπού οι πράκτορες δεν είναι επίμονοι. Μπορούμε να δούμε ότι η εύρεση τελικής γνώμης είναι πολύ γρήγορη.

|  |  |
| --- | --- |
|  |  |
|  |  |

### Τελικές παρατηρήσεις

Δύο βασικές παρατηρήσεις μπορούν να γίνουν πάνω στα δεδομένα του μοντέλου που πήραμε. Η πρώτη είναι ότι όσο ποιο πολύ επιμονή έχουν στην αρχική τους γνώμη οι πράκτορες στο μοντέλο, τόσο πιο γρήγορα συγκλίνουν οι πράκτορες σε μία τελική απόφαση. Η δεύτερη είναι ότι η επιμονή των πρακτόρων οδηγεί στο να υπάρχουν περισσότερες τιμές στις μεσαίες τιμές του εύρους μεταξύ του μηδέν και ένα.

Το πρώτο γεγονός συμβαίνει είναι ότι στο De Groot μοντέλο, το οποίο είναι το Friedkin-Johnsen Model χωρίς επιμονή αρχικής γνώμης, οι πράκτορες τροφοδοτούν τις γνώμες τους και τροφοδοτούνται με τις γνώμες των άλλων συνέχεια. Έτσι, υπάρχει μία συνεχή αλλαγής γνώμης των πρακτόρων και μόνο τα πάνω και κάτω όρια θα σταματήσουν τις επαναλήψεις. Αντίθετα, εάν υπάρχει επιμονή στην αρχική γνώμη, η γνώμη ενός πράκτορα θα παραμείνει η ίδια, οπότε θα κάνει τους πράκτορες που επηρεάζει να παραμείνουν στην ίδια άποψη επίσης. Ακόμα και σε μερική επιμονή, θα είναι πολύ πιο δυνατή κατά μέσο όρο από απόψεις άλλων πρακτόρων.

Εδώ μπορούμε να εντοπίσουμε τον τύπο , με transition matrix A και θ επιμονή των πρακτόρων οπού ναι μεν υποδεικνύει σταθερότητα, αλλά από τα πειράματα του μοντέλου μπορούμε να δούμε ότι η επιμονή επηρεάζει την ταχύτητα σύγκλισης, άρα όσο πιο μικρό το τόσο καλύτερο.

Για τις μέσες τιμές, το Friedkin-Johnsen Model συγκλίνει σε κατάσταση

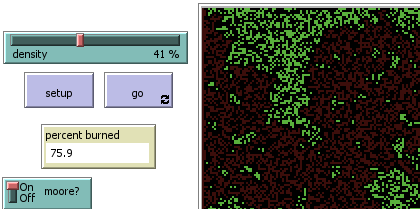
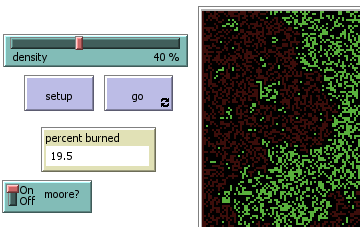
Με V να είναι:

## Άσκηση 2

Μας ζητήθηκε με βάση τον κώδικα στο έτοιμο μοντέλο της Netlogo να υλοποιήσουμε διάφορες εκδοχές του μοντέλου της φωτιάς στο δάσος. Ακολουθούν τα υποερωτήματα με τις παρατηρήσεις μας.

### Υποερώτημα 1)

Για την υλοποίηση της γειτονιάς Moore δεν χρειάστηκαν ιδιαίτερες αλλαγές, απλώς μια boolean μεταβλητή που θα ορίζει αν η γειτονιά στο μοντέλο θα είναι Moore/Von-Neuman. Έπειτα από την υλοποίηση και τον πειραματισμό με διάφορα “τρεξίματα” του μοντέλου, παρατηρήσαμε ότι η κρίσιμη πυκνότητα στο κανονικό μοντέλο με γειτονιά Moore είναι 40%, αφού στο 40% το καμένο δάσος σπάνια ήταν πάνω από το 20% στο τέλος ενώ στο 41% αυτό το ποσοστό ανέβαινε δραματικά πολύ, συστηματικά πάνω από 50%:

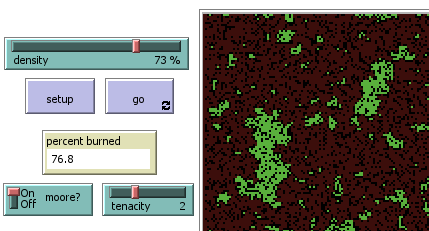


### Υποερώτημα 2)

Στην περίπτωση που για να καει ένα δέντρο χρειάζεται πάνω από ένα δέντρο γειτονικό του να καίγεται, δημιουργήσαμε μία νέα μεταβλητή με το όνομα “tenacity” η οποία αντιπροσωπεύει την αντοχή των δέντρων στην φωτιά, δηλαδή πόσα δέντρα πρέπει να καίγονται για να πιάσει ένα δέντρο φωτιά. Αυτό έχει σημασία μόνο στην γειτονιά Moore καθώς στην γειτονία Von-Neuman δεν θα πιάσει ποτέ κάποιο δέντρο φωτιά με tenacity > 1.

Η κρίσημη πυκνότητα για tenacity = 2 είναι γύρω στο 72%:

Εικόνα που περιέχει στιγμιότυπο οθόνης, κείμενο, σχεδίαση

Περιγραφή που δημιουργήθηκε αυτόματα

Για tenacity = 3 η πυκνότητα αυτή είναι περίπου κοντά στο 100% αφού για να πιάσει φωτια ένα δέντρο πρέπει 3 διπλανά του να εχουν φωτιά , κάτι πολύ περιορισμένο που με το παραμικρό κενό στην γραμμή της φωτιάς δεν επιτρέπει την μετάδοση της.

### Υποερώτημα 3)

Για την υλοποίηση του ανέμου, υποθέτουμε ότι η ένταση του αέρα θα είναι πόσα δέντρα μακριά θα μεταφερθεί η φλόγα και ως στοχαστική διαδικασία έχουμε μία πιθανόητα αυτή η φλόγα να επηρεάσει όντως τα επόμενα δέντρα. Αν αυτή η πιθανότητα είναι 0, είναι σαν να μην έχουμε αέρα ενώ αν είναι 100, τότε ο αέρας έχει πλήρη ισχύ. Επειδή υποθέτουμε ότι ο αέρας παέι δεξιά, τα patches που θα πιάσουν φωτιά θα είναι στην ουσία όλοι οι γείτονες του ακριβώς δεξιά patch ή και των επόμενων, ανάλογα την ένταση του αέρα.

Αντιστοίχως, ο αέρας στην γειτονιά Von-Neuman μεταφέρει την φωτιά σε επόμενα δέντρα αλλά μόνο στην ευθεία του δεξιά γείτονά του ακολουθώντας την μορφή της von Neuman γειτονιάς. Προφανώς αν στα επόμενα trees υπάρχουν πάνω ή κάτω δέντρα κι αυτά θα πιάσουν φωτιά, απλά ένα βήμα αργότερα.

Στα πειράματα που τρέξαμε, η γειτονιά ήταν Moore και η πιθανότητα μετάδοσης της φωτιάς μέσω του αέρα 100% ώστε να φανούν πιο ξεκάθαρα τα αποτελέσματα της διάδοσης ανάμεσα στις 2 πυκνότητες. Γενικά παρατηρήσαμε ότι όσο πιο δυνατός ο άνεμος, τόσο πιο μεγάλη η εξάπλωση της φωτιάς και πιο χαμηλή η κρίσιμη πυκνότητα. Την μεγαλύτερη διαφορά μεταξύ των ανέμων την είδαμε στην 25% πυκνότητα του δάσους.

Εικόνα που περιέχει κείμενο, γραμμή, στιγμιότυπο οθόνης, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

### Υποερώτημα 4)

Η πιθανότητα στην γειτονιά von neuman να περάσει η φωτιά στο δίπλα δέντρο προσομοιώθηκε με μία πιθανότητα q. Κατά τα πειράματα παρατηρήθηκε ότι και σε αυτήν την μετρική υπήρχε μία κρίσιμη πιθανότητα μετά την οποία η φωτιά γινόταν καταστροφική! Αν και ανεβάσαμε την πυκνότητα του δάσους στο 70%, αυτή η πιθανότητα όσο ήταν κάτω από το 90% δεν επέτρεπε την καύση πολλών δέντρων, ενώ όταν έφτασε στο 90% έκανε ραγδαία άνοδο.

Εικόνα που περιέχει κείμενο, διάγραμμα, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

Αυτό διαισθητικά συμβαίνει αφού εξισορροπείται λίγο το ποσοστό που καίγεται στο τέλος με την αρχική πυκνότητα και την πιθανότητα διάδοσης από ένα δέντρο. Αυτές οι τιμές είναι αντιστρόφως ανάλογες μεταξύ τους όπως φαίνεται από τα πειράματα, όσο μεγαλύτερη η πιθανότητα διάδοσης της φωτιάς, τόσο πιο μικρή κρίσιμη πυκνότητα πρέπει να έχει το δάσος για να καεί σε μεγάλο βαθμό και αντίστροφα!

### Υποερώτημα 5)

Υποπτευόμαστε ότι και σε αυτήν την περίπτωση το ποσοστό των βραδύκαυστων δέντρων θα είναι αντιστρόφως ανάλογο της πιθανότητας μετάδοσης της φωτιάς για μία δεδομένη κρίσιμη πυκνότητα του δάσους. Όντως μέσα από τα πειράματα

Εικόνα που περιέχει κείμενο, γραμμή, στιγμιότυπο οθόνης, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Για την εξαγωγή των δεδομένων τρέξαμε 5 φορές το μοντέλο για κάθε ζευγάρι τιμών. Όπως βλέπουμε, για σταθερή πυκνότητα του δάσους, η πιθανότητα μετάδοσης της φωτιάς στην οποία το δάσος καίγεται περισσότερο μειώνεται όσο η πυκνότητα των βραδύκαυστων δέντρων αυξάνεται. Άρα όντως είναι αντιστρόφως ανάλογη.

### Υποερώτημα 6)

Μετά την εφαρμογή του δρόμου, τα αποτελέσματα που είδαμε ήταν παρόμοια με της προηγούμενης περίπτωσης. Όσο είχαμε αέρα με μεγάλη πιθανότητα μετάδοσης, ο δρόμος δεν φάνηκε να εμποδίζει πολύ την εξάπλοση της φωτιάς, ενώ αν δεν είχαμε αέρα ή έιχε χαμηλή πιθανότητα μετάδοσης, έκανε διαφορά αφού σταματούσε την φλόγα στο πάνω μέρος του δάσους.

### Υποερώτημα 7)

Με τον συνδυασμό όλων των παραμέτρων που προσθέσαμε, παρατηρούμε πολύ διαφορετικές συμπεριφορές ως προς το τελικό ποσοστό του δάσους που καίγεται. Γενικά σαν κρίσιμη πιθανότητα **P** εδώ αναφερόμαστε στο ποσοστό πυκνότητας του δάσους πάνω από το οποίο για συγκεκριμένες τιμές των παραμέτρων το δάσος καίγεται σε τεράστιο βαθμό (στο αρχικό μοντέλο αυτή είναι το 58%)

* Για γειτονιά Moore στο αρχικό μοντέλο η P = 40%
* Για γειτονιά Moore με q = 90%, P = 44%
* Για γειτονιά Moore με q = 80%, P = 49%
* Για γειτονιά Moore με q = 80% και slow burning trees = 25%, P = 45%
* Για γειτονιά Moore με wind-intensity = 2 και active-wind = 100%, P = 28%
* Στο ίδιο για active-wind = 50%, P = 33%
* Για γειτονιά Moore, wind-intensity = 3 και active-wind = 100%, P = 21%
* Για γειτονιά Moore, wind-intensity = 3 και active-wind = 50%, P = 28%
* Για γειτονιά Von-Neuman με sbt = 25% και q = 80%, P = 69%
* Για γειτονιά Von-Neuman με sbt = 50% και q = 80%, P = 65%
* Για γειτονιά Von-Neuman με active-wind = 100%, q = 100% και wind-intensity = 2, P = 46%
* Για γειτονιά Von-Neuman με active-wind = 100%, q = 100% και wind-intensity = 3, P = 39%
* Για γειτονιά Von-Neuman με active-wind = 50%, q = 100% και wind-intensity = 2, P = 52%
* Για γειτονιά Von-Neuman με active-wind = 50%, q = 100% και wind-intensity = 3, P = 46%

Γενικά παρατηρούμε ότι για διάφορες συνθήκες η P αλλάζει τιμές. Γενικά στην γειτονιά Moore είναι πολυ πιο χαμηλή η τιμή της από την γειτονιά Von-Neuman, πράγμα λογικό αφού ένα δέντρο επηρεάζει πολλά περισσότερα patches. Επίσης βλέπουμε ότι η πιθανότητα q επηρεάζει πολύ την P όσο μειώνεται, άρα είναι αντιστρόφως ανάλογής της, όπως και το tenacity των δέντρων στην φωτιά και το active-wind % που δείχνει ποια η πιθανότητα ο αέρας να μεταφέρει την φωτιά.

Από την άλλη τα sbt (slow-burning-trees), που είναι το ποσοστό των βραδύκαυστων δέντρων σε σχέση με όλο το δάσος, όσο αυξάνεται δεν κάνει καμία διαφορά όσο οι στοχαστικές διαδικασίες είναι στο 100% (q, active-wind). Όμως αν μειώσουμε αυτά τα ποσοστά, το πλήθος των βραδύκαυστων δέντρων επηρεαζουν το τελικό αποτέλεσμα αφού αυτά φλέγονται για έναν δεύτερο γύρο και άρα δίνουν 2η ευκαιρία σε αυτά τα ποσοστά να πετύχουν και να περάσουν την φωτιά.

Ακολουθεί screenshot με το γραφικό περιβάλλον της υλοποίησης και ο κώδικας σε netlogo:

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, λογισμικό, λογισμικό γραφικών

Περιγραφή που δημιουργήθηκε αυτόματα

# Κώδικας

## Άσκηση 1

;;;in-Johnsen Model where each agent has a predisposition over his starting opinions, based of a starting stubborness value  
;;;We expect to see the model converge to a specific set of opinions.  
;;;The transition matrix visualize the influence each agent has at every other agent  
;;;For the transition matrix, we consider that each agent has 3 kind of relationships and each member of each group affect him in the same way. This implemantion was made in order to assign  
;;;random values to each element of the transition matrix,while ensuring that the product of each row will be one  
  
  
**globals** [  
 float-precision ;;To limit the amount of decimal points that are being used at the calculations  
 thereshold ;; if every turtle has change of opinion lower than the value of thereshold given, then the algorithm stops and write the final opinions  
 upper-bound ;; it has the upper bound of the opinion being given  
 having-already-written-final-results ;we want to write full results of changing the opinions at the output window, without printing it infinity times  
 median-of-opinions ;the median of the opinions  
 variance-of-opinions ;the variance of the opinions  
  
]  
  
**turtles-own** [  
  
 starting-opinions ; a list of two elements that has the two starting opinions  
 opinions ; a list of two elements that has the two current opinions  
  
 stubbornness ; how much the agent instists on the starting opinion  
 transition-row ; its part of the transition matrix  
  
 change-of-opinion ; the difference of the current opinion of the agent to its opinion at the previous iteration  
  
]  
  
;;;;;;;;;;;;;;;;;;  
;;;;;;SETUP;;;;;;;  
;;;;;;;;;;;;;;;;;;  
  
**to** setup  
 clear-all  
  
 set float-precision 5  
 set upper-bound 1 + 10 ^ (-3) ;we want to have a value a bit over 1,so an agent can have an opinion of 1. We set it later to 1  
 set thereshold 1 \* 10 ^ (- (float-precision - 1)) ;the value we check if the change of opinion is under 1 \* 10^(-4)  
  
create-turtles number-of-agents [  
  
 set shape "person" ;as opinions dynamics usually represent how each person's opinion get affected by the other people  
 ;random-float will surely output a number equal or more to zero,but it will be surely be less than the upper bound, so this is why we increase the upper bound a bit over 1  
 set starting-opinions list normalize-number (precision random-float upper-bound float-precision) normalize-number (precision random-float upper-bound float-precision)  
  
 set opinions starting-opinions  
 set transition-row n-values count turtles [0]  
 set label list agent-id opinions  
]  
 layout-circle turtles 11  
  
 set-starting-stubbornness ; we set the stubborness based one of the 4 configuration  
 ask turtles [assign-transition-row] ; each turtle builds connections to other turtles, even itselfs  
 print-transition-rows ; print the transition matrix at the output windows  
 calculate-median-and-variance  
  
 reset-ticks  
  
**end**  
  
;;;;;;;;;;;;;;;;;;  
;;;;;;GO;;;;;;;;;;  
;;;;;;;;;;;;;;;;;;  
  
  
  
**to** go  
  
 ask turtles [  
 ;we implement the Friedkin-Johnsen Model equation for getting the next opinion.We run the equation for opinion 1 and 2 seperately  
 let new-opinions list 0 0  
  
 let turtle-counter 0  
 ;we sum the opinions of each agent multiplied by the effect the have on the current agent based of its transition row  
 foreach transition-row [[the-transition-element] ->  
 set new-opinions replace-item 0 new-opinions precision (item 0 new-opinions + (item 0 [opinions] of turtle turtle-counter) \* the-transition-element) float-precision  
 set turtle-counter turtle-counter + 1  
 ]  
 ;we take the product we found previously and we insert the stubborness parameter into it  
 set new-opinions replace-item 0 new-opinions (((1 - stubbornness) \* item 0 new-opinions) + stubbornness \* item 0 starting-opinions)  
  
 set turtle-counter 0  
 ;we sum the opinions of each agent multiplied by the effect the have on the current agent based of its transition row  
 foreach transition-row [[the-transition-element] ->  
 set new-opinions replace-item 1 new-opinions precision (item 1 new-opinions + (item 1 [opinions] of turtle turtle-counter) \* the-transition-element) float-precision  
 set turtle-counter turtle-counter + 1  
 ]  
 ;we take the product we found previously and we insert the stubborness parameter into it  
 set new-opinions replace-item 1 new-opinions (((1 - stubbornness) \* item 1 new-opinions) + stubbornness \* item 1 starting-opinions)  
  
 ;we keep the result over 0 and under 1 for both opinions  
 set new-opinions replace-item 0 new-opinions ( normalize-number (precision (item 0 new-opinions) float-precision))  
 set new-opinions replace-item 1 new-opinions ( normalize-number (precision (item 1 new-opinions) float-precision))  
 ;we compare the news and olds current opinions using euclidian distance before saving the new opinions as the current ones  
 set change-of-opinion sqrt( (item 0 new-opinions - item 0 opinions)^ 2 + (item 1 new-opinions - item 1 opinions ) ^ 2 )  
 set opinions replace-item 0 opinions item 0 new-opinions  
 set opinions replace-item 1 opinions item 1 new-opinions  
  
 set label list agent-id opinions  
 ]  
 calculate-median-and-variance  
  
 tick ;we consider the upper part the algorithm and the lower back the stop condition  
  
 if all? turtles [change-of-opinion < thereshold] [  
  
  
 ;we print the first and final opinions of each agent  
 if having-already-written-final-results != **true** [  
 set having-already-written-final-results **true**  
 output-print "starting and final opinions"  
 let turtle-counter 0  
 while [turtle-counter < count turtles][  
 ask turtle turtle-counter [  
 let list-text (list "agent" turtle-counter starting-opinions "->" opinions "stubbornness" stubbornness)  
 output-print list-text  
 ]  
 set turtle-counter turtle-counter + 1  
 ]  
 ]  
 user-message "The agents have settled on their opinions"  
 stop  
 ]  
  
**end**  
  
  
  
**to** set-starting-stubbornness  
  
 if stubbornness-condition = "no-one stubborn"[  
 ask turtles [ set stubbornness 0]  
 ]  
 if stubbornness-condition = "all stubborn"[  
 ask turtles [set stubbornness 1]  
 ]  
 if stubbornness-condition = "one stubborn"[  
 ;ask turtles take turtles in random order, so by taking the first turtle we achieve randomly selecting the stubborn agent  
 let one-stubborn-guy **false**  
 ask turtles [  
 ifelse one-stubborn-guy = **false** [  
 set stubbornness 1  
 set one-stubborn-guy **true**  
 ]  
 [  
 set stubbornness 0  
 ]  
 ]  
  
 ]  
 if stubbornness-condition = "random"[  
  
 ask turtles [set stubbornness normalize-number (precision random-float upper-bound float-precision)]  
 ]  
  
 ask turtles [set size 1 + 1 \* stubbornness]  
  
**end**  
;as in transition matrix,each rows shows the relationships of each agent to the other,here we are creating it in a turtle level each row  
;we take almost 3 groups of agents and uniformally we assign to each group a percentage of the connection the group of agents will share. We want the rows of the transition matrix to be stochastic,  
;so the product of the elements of row will results to one   
**to** assign-transition-row  
  
  
 let turtles-yet-to-be-chosen count turtles ; how many turtles we have yet shosen  
  
  
 let turtles-chosen-amount ( (random turtles-yet-to-be-chosen ) + 1) ; we increase it one more in order not to have the problem of taking a zero number  
 set turtles-yet-to-be-chosen turtles-yet-to-be-chosen - turtles-chosen-amount  
   
 assign-values-to-each-row turtles-yet-to-be-chosen turtles-chosen-amount 0.5 1  
  
 if turtles-yet-to-be-chosen > 0 [ ; we have the posibility of choosing all the elements for the first group  
   
 set turtles-chosen-amount ( (random turtles-yet-to-be-chosen ) + 1)  
 set turtles-yet-to-be-chosen turtles-yet-to-be-chosen - turtles-chosen-amount  
 assign-values-to-each-row turtles-yet-to-be-chosen turtles-chosen-amount 0.3 (1 - 0.5)  
 ]  
 if turtles-yet-to-be-chosen > 0 [; we have the posibility of choosing all the elements for the second group  
   
 set turtles-chosen-amount ( (random turtles-yet-to-be-chosen ) + 1)  
 set turtles-yet-to-be-chosen turtles-yet-to-be-chosen - turtles-chosen-amount  
 assign-values-to-each-row turtles-yet-to-be-chosen turtles-chosen-amount 0.2 0.2  
 ]  
**end**  
;we have to reassure that if all the remaining turtles all chosen, they will be shared all the remaining propability instead of the given one  
**to** assign-values-to-each-row [turtles-yet-to-be-chosen turtles-chosen-amount propability-given remaining-percentage-to-be-given]  
  
 ifelse turtles-yet-to-be-chosen = 0;if the group has all the elements with zero values, we assign the rest of communication percentage we havε το γιωε  
 [assign-values-to-each-row-for-each-element turtles-chosen-amount (remaining-percentage-to-be-given / turtles-chosen-amount) ]  
 [assign-values-to-each-row-for-each-element turtles-chosen-amount (propability-given / turtles-chosen-amount)]  
  
**end**  
; we have to randomly assign elements of the vector that haven't yet given a non zero value,based of the size of zeros  
**to** assign-values-to-each-row-for-each-element [elements-to-choose propability-given]  
 set propability-given precision propability-given float-precision  
 let random-index 0  
 ;we cound use ask n of transition-row while filtering the no zero elements,but then we will not have the position of the elements at the transition rows  
 ;so we prefer the stupid method of choosing randomly from all the list  
 while [elements-to-choose > 0] [  
 set random-index random count turtles  
 if item random-index transition-row = 0  
 [  
 set transition-row replace-item (random-index) transition-row precision propability-given 3  
 set elements-to-choose elements-to-choose - 1  
 ]  
 ]  
  
**end**  
;we print the transition matrix  
**to** print-transition-rows  
  
 let turtle-counter 0  
 output-print "transition matrix"  
 output-print ""  
 while [turtle-counter < count turtles][  
  
 ask turtle turtle-counter [  
 output-print (sentence agent-id opinions "stubbornness" stubbornness)  
 output-print transition-row  
 output-print ""  
 ]  
 set turtle-counter turtle-counter + 1  
 ]  
**end**  
**to** calculate-median-and-variance  
  
 set median-of-opinions list (median [item 0 opinions] of turtles) (median [item 1 opinions] of turtles)  
 set variance-of-opinions list (4 \* variance [item 0 opinions] of turtles ) (4 \* variance [item 1 opinions] of turtles)  
  
**end**  
;we keep the number in a limit  
**to-report** normalize-number [number]  
 ( ifelse  
 number < 0 [report 0]  
 number > 1 [report 1]  
 [report number]  
 )  
**end**  
; we just use it to show the id of an agent that have the particular opinions or the transition matrixes  
**to-report** agent-id  
 report (word "agent " who ":")  
**end**

## Άσκηση 2

**globals** [  
 initial-trees ;; how many trees (green patches) we started with  
 burned-trees ;; how many have burned so far  
]  
  
**patches-own** [  
 slow? ;; bollean to set a tree as slow burning  
]  
  
**breed** [fires fire] ;; bright red turtles -- the leading edge of the fire  
**breed** [embers ember] ;; turtles gradually fading from red to near black  
  
**to** setup  
 clear-all  
 reset-ticks  
 set-default-shape turtles "square"  
 ;; make some green trees  
  
 ask patches with [(random-float 100) < density] [  
 set pcolor green  
 set slow? **false**  
  
 ;; if a tree is slow burning, it's yellow  
 if random-float 100 < slow-burning-trees [  
 set slow? **true**  
 set pcolor yellow  
 ]  
 ]  
  
 ;; make a column of burning trees  
 ask patches with [pxcor = min-pxcor]  
 [ ignite ]  
 ;; set tree counts  
 set initial-trees count patches with [pcolor = green or pcolor = yellow]  
 set burned-trees 0  
  
 ;; create a road in the middle spanning half the forest  
 if road? [  
 ask patches with [pxcor = 0 and pycor > 0 ] [  
 set pcolor white  
 ]  
 ]  
  
**end**  
  
**to** go  
 if not any? turtles ;; either fires or embers  
 [ stop ]  
  
 ask fires [  
;;;;;; with moore neighborhood, all 8 of the neighbors are ignited  
 ifelse moore? [  
  
 ;; with probability q the fire spreads from the current tree  
 if random 100 <= q [  
 ask neighbors with [pcolor = green or pcolor = yellow]  
 [ ignite ]  
  
 ;; wind transfers the fire for "wind-intensity" patches to the right  
 if wind? and random 100 < active-wind [  
  
 ;; if the wind intensity is more than 1, then set ablaze the trees to the right of  
 ;; my right neighbor, so all of its neighbors.  
 if wind-intensity >= 2 [  
 ;; safety measure for the edges of the world  
 if pxcor < max-pxcor - 2 [  
 ask patch (pxcor + 1) pycor [  
 ask neighbors with [pcolor = green or pcolor = yellow] [ force-ignite ]  
 ]  
 ]  
 ]  
  
 ;; if the wind intensity is 3, then set ablaze the trees 2 trees  
 ;; away form my neighbors  
 if wind-intensity = 3 [  
 if pxcor < max-pxcor - 3 [  
 ask patch (pxcor + 2) pycor [  
 ask neighbors with [pcolor = green or pcolor = yellow] [ force-ignite ]  
 ]  
 ]  
 ]  
 ]  
 ]  
 ]  
  
;;;;;; with von-neuman neighborhood, only 4 neighbors are considered  
 [  
 ;; with probability q the fire spreads from the current tree  
 if random 100 <= q [  
 ask neighbors4 with [pcolor = green or pcolor = yellow]  
 [ ignite ]  
  
 ;; wind transfers the fire for "wind-intensity" patches to the right  
 if wind? and random 100 < active-wind [  
  
 ;; if the wind intensity is more than 1, then set ablaze the tree one tree  
 ;; away form my neighbors  
 if wind-intensity >= 2 [  
 if pxcor < max-pxcor - 2 [  
 ask patch (pxcor + 2) pycor [  
 if pcolor = green or pcolor = yellow [ force-ignite ]  
 ]  
 ]  
 ]  
  
 ;; if the wind intensity is 3, then set ablaze the trees 2 trees  
 ;; away form my neighbors  
 if wind-intensity = 3 [  
 if pxcor < max-pxcor - 3 [  
 ask patch (pxcor + 3) pycor [  
 if pcolor = green or pcolor = yellow [ force-ignite ]  
 ]  
 ]  
 ]  
 ]  
 ]  
 ]  
  
 ;; simulate the slow burn of the trees  
 ifelse [slow?] of patch-here = **false** [  
 set color red  
 set breed embers  
 ]  
 [  
 ;; if the patch is a slow burning tree, make the flame white  
 ;; and set slow? false to make it an ember next round  
 set slow? **false**  
 set color white  
 ]  
  
 ]  
 fade-embers  
 tick  
**end**  
  
  
;; creates the fire turtles  
**to** ignite ;; patch procedure  
  
 ;; get for each patch that calls the procedure the current neighbor burning trees number  
 let ignited-neig count neighbors with [any? fires-here]  
 ;; if the ticks are 0, then setup the starting line of fire  
 if ticks = 0 [  
 force-ignite  
 ]  
 ;; if the number of ignited neighbors is less than the tenacity, don't ignite  
 if ignited-neig >= tenacity [  
 force-ignite  
 ]  
**end**  
  
  
**to** force-ignite ;; patch procedure  
 sprout-fires 1  
 [ set color red ]  
 set pcolor black  
 set burned-trees burned-trees + 1  
**end**  
  
  
;; achieve fading color effect for the fire as it burns  
**to** fade-embers  
 ask embers  
 [ set color color - 0.3 ;; make red darker  
 if color < red - 3.5 ;; are we almost at black?  
 [ set pcolor color  
 die ] ]  
**end**