

# Università degli Studi di Perugia



Intelligent Models  
Relazione del progetto *Covid Distantiation*

Mattia Polticchia  
Matricola: 332040

Matteo Baldassarrini  
Matricola: 335286

# Indice

<b>1</b>	<b>Problema ed ipotesi risolutive</b>	<b>4</b>
1.1	Problema . . . . .	4
1.2	Ipotesi Risolutive . . . . .	4
1.2.1	Persone . . . . .	4
1.2.2	Distanziatori . . . . .	5
1.2.3	Ambiente . . . . .	5
<b>2</b>	<b>Caratteristiche dello sviluppo</b>	<b>6</b>
2.1	Introduzione . . . . .	6
2.2	Ambiente . . . . .	6
2.2.1	Variabili globali . . . . .	6
2.2.2	Funzione Set-up . . . . .	7
2.2.3	Conta-fuorilegge . . . . .	8
2.2.4	Funzione controlla-tick . . . . .	8
2.3	Persone . . . . .	9
2.3.1	Parametri . . . . .	9
2.3.2	Metodi . . . . .	9
2.4	Distanziatori . . . . .	12
2.4.1	Metodi . . . . .	12
2.5	Funzione Go . . . . .	15
<b>3</b>	<b>Test effettuati e analisi dei risultati</b>	<b>16</b>
3.1	Introduzione . . . . .	16
3.2	Test sul Movimento Distanziatori . . . . .	16
3.3	Visione maggiore delle distanze minime . . . . .	17
3.4	Visione minore della propria distanza . . . . .	18
3.5	Distanza minima percepita minore della globale . . . . .	19
3.6	Distanza minima percepita maggiore della globale . . . . .	20
3.7	Distanza minima percepita uguale alla globale . . . . .	21
<b>4</b>	<b>Conclusioni</b>	<b>22</b>

# Introduzione

Questa relazione riguarda lo sviluppo del progetto *Covid Distantiation* riguardante l'argomento *Distribute Intelligence e Swarm Computing*.

Il progetto è stato sviluppato in NetLogo e al seguente link è possibile vederne il github. <https://github.com/>

## Covid Distantiation

Vi sono due tipi di agenti: le persone (numero parametrico), che si muovono in maniera random nello spazio, hanno un cono visivo (parametro) e vengono attratte convergendo in misura diversa (parametro) verso i vicini da cui mantengono una distanza minima (parametro). I distanziatori hanno un cono visivo in generale più lungo delle persone ed hanno il compito di distanziare le persone che sono ad una distanza inferiore al minimo globale (parametro). Se una persona ha un minimo corrente inferiore a quello globale e vede un distanziatore nel suo cono visivo, aumenta il proprio minimo a quello globale e lo mantiene per un certo numero di tic, dopo di che ritorna alla distanza minima delle persone. Ad ogni passo viene mostrato il numero di persone a distanza inferiore al minimo globale. Definire una strategia distribuita opportuna di comportamento dei distanziatori. Eventualmente definita da parametri per far in modo da minimizzare il numero di persone a distanza inferiore alla minima globale. Es. Autodistribuire i distanziatori o randomizzarli.

# Capitolo 1

## Problema ed ipotesi risolutive

### 1.1 Problema

A partire dal problema abbiamo riassunto i punti focali dei fatti verificatisi durante il periodo del COVID-19. È stata imposta una distanza minima di sicurezza per arginare la diffusione del virus e il contagio. Parliamo di assembramento quando due o più persone non mantengono la distanza e, spesso, si trovano ferme in un posto. Per evitarne la formazione si è dato il compito alle varie forze dell'ordine di controllare e disincentivare gli ammassamenti. Si è notato che le persone, in presenza di forze dell'ordine, tendevano a comportarsi seguendo le regole.

### 1.2 Ipotesi Risolutive

A partire da queste situazioni abbiamo modellato e creato un sistema che simulasse queste principali caratteristiche comportamentali. Inizialmente abbiamo supposto di dividere gli agenti ( *chiamati anche turtles nel seguito di questa relazione* ) in due gruppi.

- Persone
- Distanziatori

#### 1.2.1 Persone

Per le persone abbiamo deciso di assegnare dei valori che ne influenzassero il comportamento.

- **Cono visivo:** angolo e lunghezza di visione delle persone
- **Distanza Minima:** distanza necessaria percepita che si limitano a mantenere dalle altre persone che vedono
- **Convergenza:** tendenza delle persone nell'essere attratte verso le altre, girando di un certo numero di gradi verso le persone viste.

Per diversificare le persone questi valori sono stati leggermente differenziati per tutta la popolazione in modo da sembrare indipendenti.

**Comportamenti:** Il movimento è stato modellato in modo da essere casuale nell'ambiente e che in presenza di altre persone, ma non di distanziatori, le persone tendessero all'assembramento prendendo spunto dalle situazioni reali.

Un altro comportamento pensato per le persone è quello di distanziarsi tra loro qualora venissero visti dai distanziatori all'interno di un assembramento.

### 1.2.2 Distanziatori

Allo stesso modo i distanziatori hanno una serie di valori che ne influenzano la modalità di comportamento.

- **Cono visivo:** angolo e lunghezza di visione dei distanziatori  
In questo caso sarà più lungo e ampio delle persone in quanto il loro compito è di vigilare nell'ambiente
- **Movimento:** Questo parametro imposta quale tipo di "pattugliamento" dovranno effettuare

**Comportamenti:** I distanziatori si muovono seguendo il parametro e sono addetti a far rispettare la distanza imposta. Quando si trovano in presenza di un assembramento si fermano e "ordinano" alle persone di separarsi.

### 1.2.3 Ambiente

Abbiamo assegnato delle caratteristiche pure all'ambiente in quanto necessarie per simulare un "governo". Molte delle regole quali la distanza globale da mantenere e quanto le persone tendessero ad avvicinarsi sono state lasciate alla decisione dell'utente.

- Distanza Minima Globale, come già detto è la distanza che le persone devono rispettare
- Quantità di persone presenti nel modello
- Tutti i parametri trattati negli agenti

# Capitolo 2

## Caratteristiche dello sviluppo

### 2.1 Introduzione

Abbiamo sviluppato tutto il progetto nel software NetLogo che ci ha fornito parte dell'ambiente offrendoci tutto il necessario per avere un sistema di agenti. Tramite la sezione interface abbiamo dato la possibilità di vedere l'ambiente e di inserire i parametri precedentemente trattati. Tramite la sezione code abbiamo scritto l'effettivo codice degli agenti.

### 2.2 Ambiente

Abbiamo diviso le turtle in due *breed* così da essere in grado di effettuare una distinzione tra essi. I due gruppi sono rispettivamente come segue ed anche singolo agente:

```
breed [distanziatori distanziatore]  
breed [persone persona]
```

#### 2.2.1 Variabili globali

Abbiamo dichiarato delle variabili globali per poter facilitare l'ottenimento di informazioni dentro le funzioni dei rispettivi distanziatori e persone. La variabile giro, ad esempio, ci permette di

```
globals [  
  giro      ;; usata per colorare il giro di distanza delle persone  
  
  id        ;; usato per memorizzare temporaneamente l'id dei distanziatori  
  
  offset    ;; serve per il plot dei distanziatori, ne modifica la distanza di spawn  
]
```

effettuare il plot del cerchio che sta attorno alle singole persone e rappresenta la distanza percepita.

## 2.2.2 Funzione Set-up

Netlogo offre una funzione di inizializzazione chiamata `set-up` in cui abbiamo impostato i valori necessari al funzionamento del sistema. Alcuni di essi sono stati presi dagli slider utilizzati nella *interface* ed assegnati ai rispettivi agenti.

```
to setup
  clear-all

  set offset popolazione
  create-persone popolazione
  [
    set color yellow - 2 + random 7    ;; definizione estetica dell'agente
    set size 1.5
    setxy random-xcor random-ycor
    set shape "person"
    set persone-viste no-turtles        ;; -----

    set angolo-visione 120              ;; Variabili di comportamento
    set mantieni-per 0
    assegna-vista
    assegna-convergenza
    assegna-distanza-minima
    set cammina true
    set fuorilegge false                ;; -----
  ]
  create-distanziatori ( ceiling ( popolazione / 5 ) )
  [
    set color blue                      ;; definizione estetica dell'agente
    set size 2
    set shape "square"
    setxy ((offset - who) * ( world-width / 5 )) * -1 ((offset - who) * ( world-height / 5 )) * -1
    set label-color red                 ;; -----

    set mov movimenti                  ;; Variabili di comportamento
    set heading 90
    set visione visione-globale * 2
    set angolo-visione 60
    set cammina true                   ;; -----
  ]

  reset-ticks
end
```

Lo spawn delle persone è casuale nell'ambiente mentre quello dei distanziatori viene generato lungo la diagonale della piccola interfaccia di visualizzazione utilizzando `offset` per tenerli equidistanti rispetto alla grandezza della mappa.

Abbiamo deciso di far spawnare i distanziatori in rapporto al numero di persone nella popolazione, dividendo per 5 e arrotondando per eccesso.

Allo stesso tempo abbiamo assegnato alle persone l'immagine "person" offerta dall'ambiente e ai distanziatori lo "square", di colore blu, per richiamare le macchine delle forze dell'ordine.

Di seguito descriviamo le funzioni usate per assegnare alle persone i valori delle singole persone inerenti alla vista, convergenza e distanza minima.

```
to assegna-vista
  set visione random-near visione-globale
end

to assegna-convergenza
  set convergenza random-near convergenza-globale
end

to assegna-distanza-minima
  set distanza-minima random-near distanza-minima-media
  set distanza-minima-tmp distanza-minima
end
;; per le persone, assegna la distanza secondo la distribuzione normale
;; per rendere le persone un po' diverse ed abbiano una diversa concezione
;; del rischio
to-report random-near [center] ;; turtle procedure
  let result 0
  repeat 40
    [ set result (result + random-float center) ]
  report result / 20
end
```

La funzione Random-near è utilizzata per assegnare alle singole persone dei valori vicini a quelli inseriti in input dall'utente seguendo una distribuzione normale. Come già detto in fase di progettazione viene fatto per rendere le persone non tutte uguali.

### 2.2.3 Conta-fuorilegge

Il valore nell'interfaccia che indica il numero di persone che infrangono la distanza minima globale, è calcolato tramite la variabile *conta-fuorilegge*.

```
to-report conta-fuorilegge
  ifelse any? persone
    [report count persone with [count other persone in-radius distanza-minima-globale > 0 ] ]
    [report 0]
end
```

Il calcolo viene effettuato contando le persone che all'interno del loro raggio di distanza minima globale hanno almeno un'altra persona. Nel caso in cui non ci sono persone generate, il valore di default mostrato sarà di 0.

### 2.2.4 Funzione controlla-tick

Questa funzione è utilizzata per controllare i tick che devono passare per far mantenere alle persone la distanza minima globale.

```
to controlla-tick
  ifelse mantieni-per = 0
    [set distanza-minima distanza-minima-tmp]
    [set mantieni-per mantieni-per - 1]
end
```



Inizialmente viene controllato se il valore di *mantieni-per* è uguale a 0, nel caso lo sia ristabilisce la distanza minima della persona a quella impostata inizialmente. Altrimenti il suo valore viene diminuito di una unità.

Nelle sezioni seguenti tratteremo le parti focali del codice utilizzato per gli agenti andando a mostrare come sono stati implementati i parametri e i comportamenti.

## 2.3 Persone

### 2.3.1 Parametri

Di seguito riportiamo i parametri delle persone con una breve spiegazione vicino.

```
persone-own [  
  persone-viste      ;; insieme delle persone viste  
  vicino-piu-vicino  ;; persona vista più vicina  
  distanziatori-visti ;; insieme dei distanziatori visti  
  fuorilegge         ;; indica se la persona infrange la legge non rispettando la distanza minima globale  
  
  persone-vicino     ;; insieme delle persone vicine nel raggio di distanza minima  
  cammina            ;; usato come flag per il movimento. Se falso, la persona si ferma  
  
  visione            ;; lunghezza in pixel della vista  
  angolo-visione     ;; gradi del cono visivo  
  
  convergenza        ;; valore che rappresenta la tendenza della persona nel convergere verso altre persone  
  distanza-minima    ;; distanza minima rispettata tra le persone  
  
  distanza-minima-tmp ;; usata per memorizzare la distanza minima quando viene sostituita  
  mantieni-per       ;; numero dei tick trascorsi prima di cambiare da distanza minima globale alla propria distanza  
]
```

Vogliamo porre una nota di attenzione sui parametri: *fuorilegge* e *distanziatori-visti*.

Il primo parametro è stato pensato per simulare la consapevolezza della persona di infrangere la legge trovandosi in un assembramento. In caso sia **True** la persona è fuorilegge mentre **False** nel caso non lo sia.

Distanziatori-visti viene utilizzata per impostare il senso di rispetto delle leggi che si verifica quando le persone vedono i distanziatori aumentando, di conseguenza, la distanza minima percepita a quella imposta dalla legge.

### 2.3.2 Metodi

In questa sezione, andremo a descrivere i diversi metodi applicati alle persone per influenzarne in comportamento.

#### Funzione osserva

Il primo metodo che andremo a spiegare è *osserva*, utilizzato dalle persone per verificare se nel loro campo visivo sono presenti altre persone e di conseguenza, in base alla propria distanza minima, decidere se separarsi, dirigersi verso di essa e fermarsi.

```

;; utilizzata dalle persone per decidere se separarsi o avvicinarsi e fermarsi
to osserva
  trova-persone-viste
  ifelse any? persone-viste
    [ trova-piu-vicino
      ifelse distance vicino-piu-vicino < distanza-minima
        [ separati
          set cammina true ]
        [ if cammina [convergi]
          set cammina false ]
      ]
    [set cammina true]
  end

;; assegna alla variabile persone-viste l'insieme delle persone presenti nel raggio visivo
to trova-persone-viste

  set persone-viste other persone in-cone visione angolo-visione
end

;; trova la persona piu vicina tra quelle viste
to trova-piu-vicino

  set vicino-piu-vicino min-one-of persone-viste [distance myself]
end

```

Al suo interno viene utilizzata la funzione *trova-persone-viste* per prendere in considerazione le persone presenti nel raggio visivo. Successivamente, se è presente almeno una persona, viene richiamata la funzione *trova-piu-vicino* che tramite la variabile *vicino-piu-vicino* assegna alla persona che ha visto la persona più vicina.

Se è stato rivelato un vicino e la distanza è minore della distanza minima percepita allora si separeranno tramite *separati*. Al contrario, se si vedono ma sono lontani la persona che vede tenderà a convergere verso quella più vicina vista.

Per le funzioni *separati* e *convergi* abbiamo fatto riferimento al modello dei **flocking** presente tra quelli offerti da Netlogo in quanto una parte del comportamento dei flock assomigliava al comportamento delle persone da noi sviluppate.

```

;; SEPARAZIONE
to separati
  turn-away ([heading] of vicino-piu-vicino) 90
end

;; CONVERGENZA
to convergi
  turn-towards average-heading-towards-persone-viste convergenza
end

```

## Funzione vedi-distanziatori

Questa funzione è utilizzata per permettere di vedere i distanziatori all'interno del proprio campo visivo alle persone. Nel caso in cui venga visto almeno un distanziatore e la distanza minima è inferiore a quella globale, le persone si metteranno automaticamente in allerta aumentado, così, la distanza minima percepita per un certo numero di tick.

```
;;cerca i distanziatori impostando di conseguenza la distanza-minima da mantenere
to vedi-distanziatori

trova-distanziatori
if any? distanziatori-visti
[
  if distanza-minima < distanza-minima-globale
  [set distanza-minima distanza-minima-globale
   set mantieni-per 100]
]
]
end

;;trova l'insieme dei distanziatori visti dalla persona
to trova-distanziatori
set distanziatori-visti other distanziatori in-cone visione angolo-visione
end
```

Al suo interno è utilizzata la funzione *trova-distanziatori* che imposta nell'agent-set *distanziatori-visti* l'insieme dei distanziatori trovati nel raggio visivo della persona.

## Funzione imposta-fuorilegge

Serve ad indicare se la persona è fuorilegge o meno sulla base delle persone che si trovano ad una distanza inferiore a quella globale rispetto ad essa.

```
;; Le persone hanno consapevolezza del fatto che stiano infrangendo la legge
to imposta-fuorilegge
trova-persone-intorno
ifelse any? persone-vicino
[set fuorilegge true]
[set fuorilegge false]
end
```

Viene utilizzata come funzione ausiliaria *trova-persone-intorno*, che cerca le persone situate nel raggio di lunghezza pari a quella minima globale, e assegnando l'agent-set trovato alla variabile *persone-vicino*.

## 2.4 Distanziatori

Per i distanziatori i parametri all'interno del codice sono stati dichiarati nel seguente modo:

```
distanziatori-own [  
  persone-viste      ;; insieme delle persone viste  
  
  visione            ;; lunghezza in pixel della vista  
  angolo-visione     ;; gradi del cono visivo  
  
  cammina           ;; usato come flag per il movimento. Se falso, il distanziatore si ferma  
  
  mov                ;; tipo di movimento selezionato del distanziatore  
]
```

Rispetto alle persone, il parametro che identifica la lunghezza della *visione* è uguale per tutti i distanziatori. *persone-viste* viene utilizzato per tenere conto delle persone che si stanno vedendo in quel tick e se sono fuorilegge andrà ad intervenire sull'assembramento in corso.

### 2.4.1 Metodi

In questa sezione descriveremo le funzioni utilizzate dai distanziatori per operare all'interno del mondo.

#### Funzione muovi

Questa funzione definisce il movimento che i distanziatori dovranno seguire in base alla variabile *mov* inizializzata precedentemente nella funzione *set-up* tramite il chooser presente nell'interfaccia. Così da poter avere più tipologie di movimento.

```
to muovi  
  (ifelse  
    mov = "randomized" [  
      if ticks mod ( offset * 2 ) = 0  
      [  
        rt random-float 360  
      ]  
      fd 0.5  
    ]  
    mov = "squared" [  
      if ticks mod ( offset * 2 ) = 0  
      [  
        rt 90  
      ]  
      fd 0.5  
    ]  
    mov = "lined" [  
      fd 0.5  
    ]  
  ]  
end
```

I movimenti definiti sono 3:

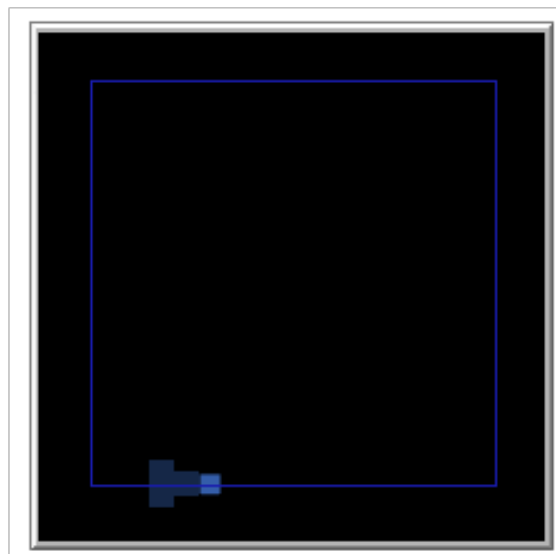
Tutti e tre i movimenti sfruttando *fd* per muoversi in avanti e *rt* per girare di x gradi. La rotazione viene effettuata, sfruttando la variabile *offset*, dopo un determinato numero di ticks.

- Random effettua una rotazione casuale dopo ogni tot di ticks
- Squared dove la rotazione è di 90° verso destra
- Lined proseguono solo dritti nell'ambiente

Di seguito riportiamo una rappresentazione grafica dei movimeni:



Movimento lined del distanziatore



Movimento squared del distanziatore

## Funzione vigila

Inizialmente, l'agente, crea un personale agent-set di persone viste partendo da quelle presenti nel proprio cono visivo dopodiché assegnamo l'identificatore del vigile alla variabile *vigile*. In seguito andiamo a controllare se l'agent-set del vigile è popolato. Nel caso lo sia, controlliamo se le persone sono fuorilegge.

```
;; controlla le persone viste, se infrangono la legge manterranno la distanza-minima-globale
to vigila
  ;; assegno le persone viste alla variabile
  set persone-viste other persone in-cone visione angolo-visione
  let vigile distanziatore who
  ifelse any? persone-viste [
    ask persone in-cone visione angolo-visione
    [
      ifelse fuorilegge
      [
        ask vigile [set cammina false set label "!"]
        if distanza-minima < distanza-minima-globale
        [set distanza-minima distanza-minima-globale
          set mantieni-per 100
        ]
        if not cammina
        [set heading ([towards myself + 180] of min-one-of persone-vicino [distance myself])]
      ]
      [
        ask vigile [set cammina true set label "" ]
      ]
    ]
  ]
  [set cammina true set label ""]
end

to vedi-persone
  set persone-viste other persone in-cone visione angolo-visione
end
```

All'occorrenza di fuorilegge nel campo visivo il vigile si ferma, farà apparire un "!" rosso per simulare lo stato di allerta e di conseguenza inciterà le persone ad allontanarsi.

## 2.5 Funzione Go

Per come è strutturato NetLogo la funzione go viene eseguita ogni tick. Noi qui abbiamo inserito tutte le funzioni principali del modello. Abbiamo diviso in tre sezioni, quella delle persone, quella dei distanziatori e quella in cui effettuiamo il render. Principalmnte abbiamo le funzioni che permettono di camminare e quelle che permettono agli agenti di interagire con l'ambiente.

```
to go
  clear-patches

  ask persone [ osserva ]
  ask persone [ imposta-fuorilegge ]
  ask persone [ controlla-tick ]
  ask persone [ vedi-distanziatori ]
  ask persone [ if cammina [ fd 0.5 ] ]   ;;se cammina è vero, la persona continua a camminare

  ;;ask persone [controlla-sliders]

;; azioni
  ask distanziatori [
    ;; applico la regola vigila
    vigila
  ]

  ask distanziatori [ if cammina [ muovi ] ]   ;;non cammina quando vede fuorilegge

  ;; inizio render -----
  ask distanziatori [
    ask patches in-cone visione angolo-visione [set pcolor blue - 3]
  ]

  ask persone [ colora-distanza ]
  ask persone [ colora-vista ]
  display
  ;; fine render -----

  tick
end
```

*Le funzioni di render sono state omesse inquanto non inerenti al funzionamento del modello, sono comunque reperibili all'interno del programma.*

# Capitolo 3

## Test effettuati e analisi dei risultati

### 3.1 Introduzione

Una volta completato lo sviluppo del modello abbiamo eseguito dei test andando a modificare i parametri. Abbiamo focalizzato le osservazioni sul formarsi degli assembramenti e valutando la bontà del parametro selezionato tramite la quantità e la frequenza di assembramenti. Abbiamo diviso i test in alcune macrocategorie alcuni per tipologia di movimento dei distanziatori mentre altri per tipologia di parametri inseriti, in modo da coprire più casi di studio possibili.

Dato che l'esecuzione prosegue nel tempo ed è difficile riportarla nella relazione daremo alcune rappresentazioni di situazioni e faremo riferimento a quelle nella descrizione degli avvenimenti durante i test.

*I seguenti test sono stati effettuati tutti con una popolazione di 30 persone e lo screenshot è stato fatto dopo, circa, 600 ticks.*

### 3.2 Test sul Movimento Distanziatori

- **Movimento Lined**

Gli assembramenti si verificano nei luoghi non toccati dalle linee e dalla visione dei distanziatori. Più è ampio il raggio di visione più gli assembramenti saranno di minore dimensione in quanto l'area coperta è maggiore.

- **Movimento Squared**

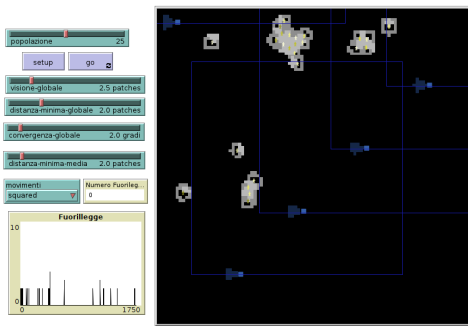
Gli assembramenti si verificano nelle aree interne ed esterne dei quadrati in quanto i distanziatori muovendosi sempre lungo quel percorso non riescono a raggiungere i luoghi fuori di esso.

Rispetto il precedente metodo viene comunque controllata più area e lascia gli assembramenti in minor numero di punti.

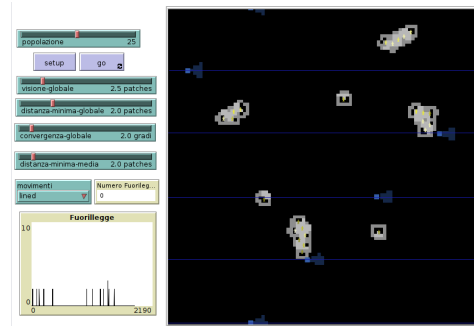
- **Movimento random**

È il più prestante perché prima o poi coprirà tutte le aree e può evitare la formazione di assembramenti in luoghi non pattugliati come nei precedenti metodi di movimento.





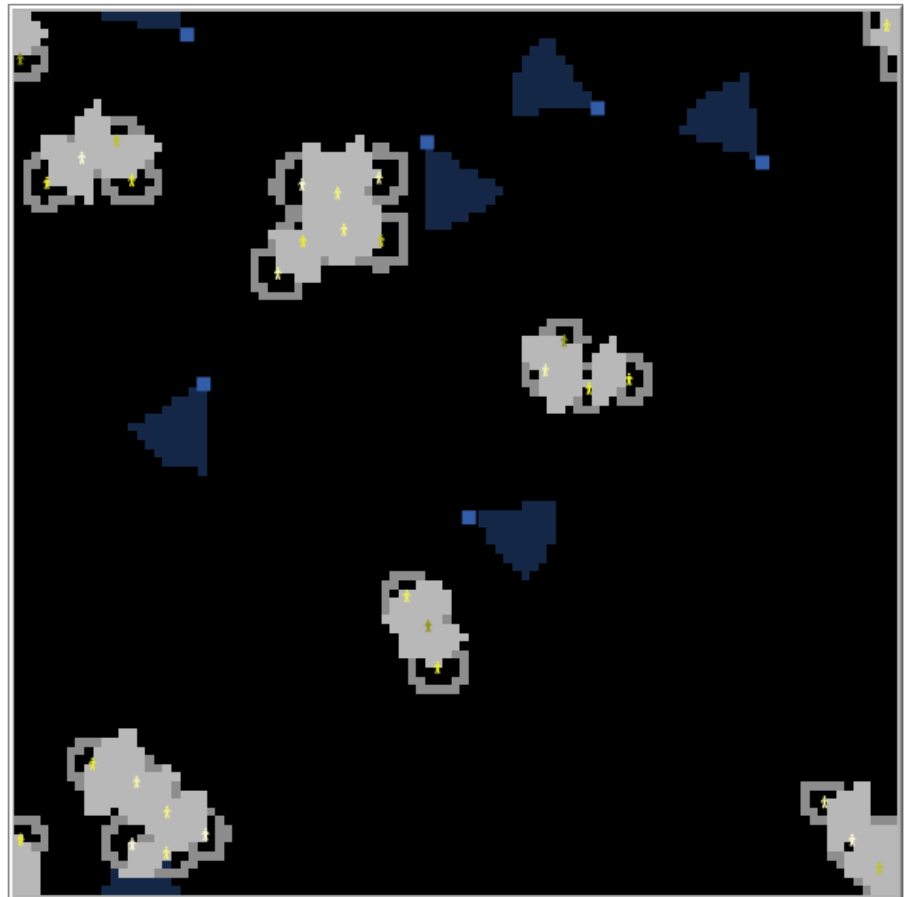
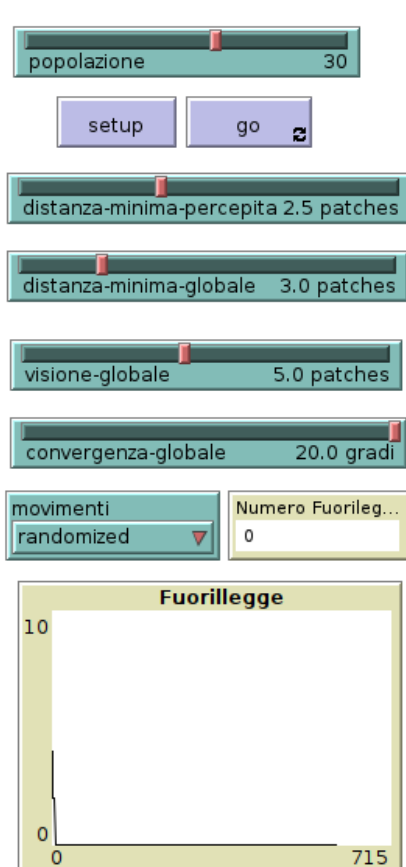
Movimento squared



Movimento lined

### 3.3 Visione maggiore delle distanze minime

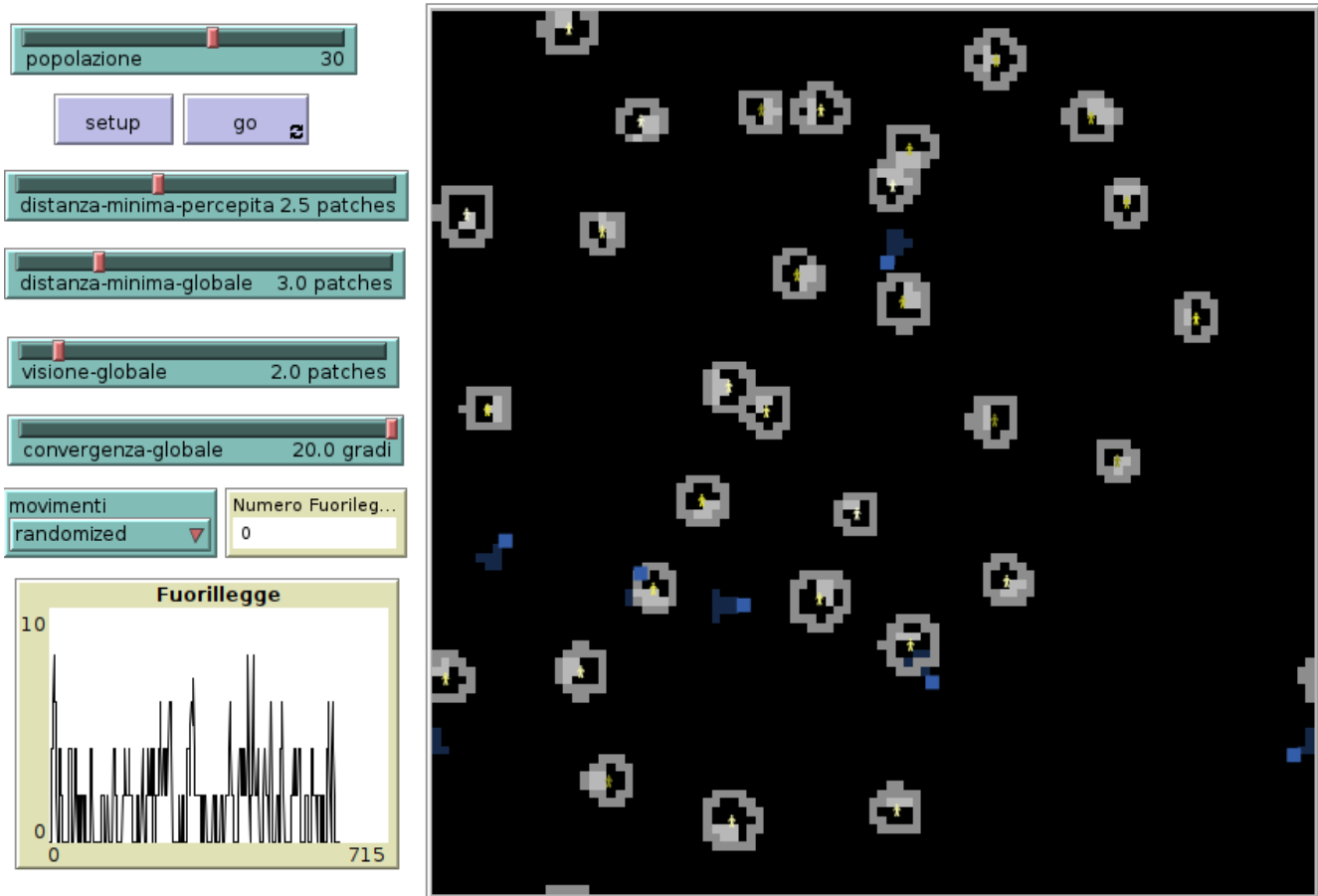
Dalla simulazione effettuata con i parametri indicati è emerso che con una lunghezza del cono visivo maggiore rispetto alla distanza minima percepita, viene favorita la creazione di assembramenti.



Si è notato anche che se questa lunghezza supera anche la distanza minima globale, nonostante si formino assembramenti, sono comunque ad una giusta distanza di sicurezza e saranno pochi i casi in cui qualcuno infranga la legge. Di conseguenza i distanziatori passeranno attraverso senza fermarsi perché non viene infranta alcuna distanza.

### 3.4 Visione minore della propria distanza

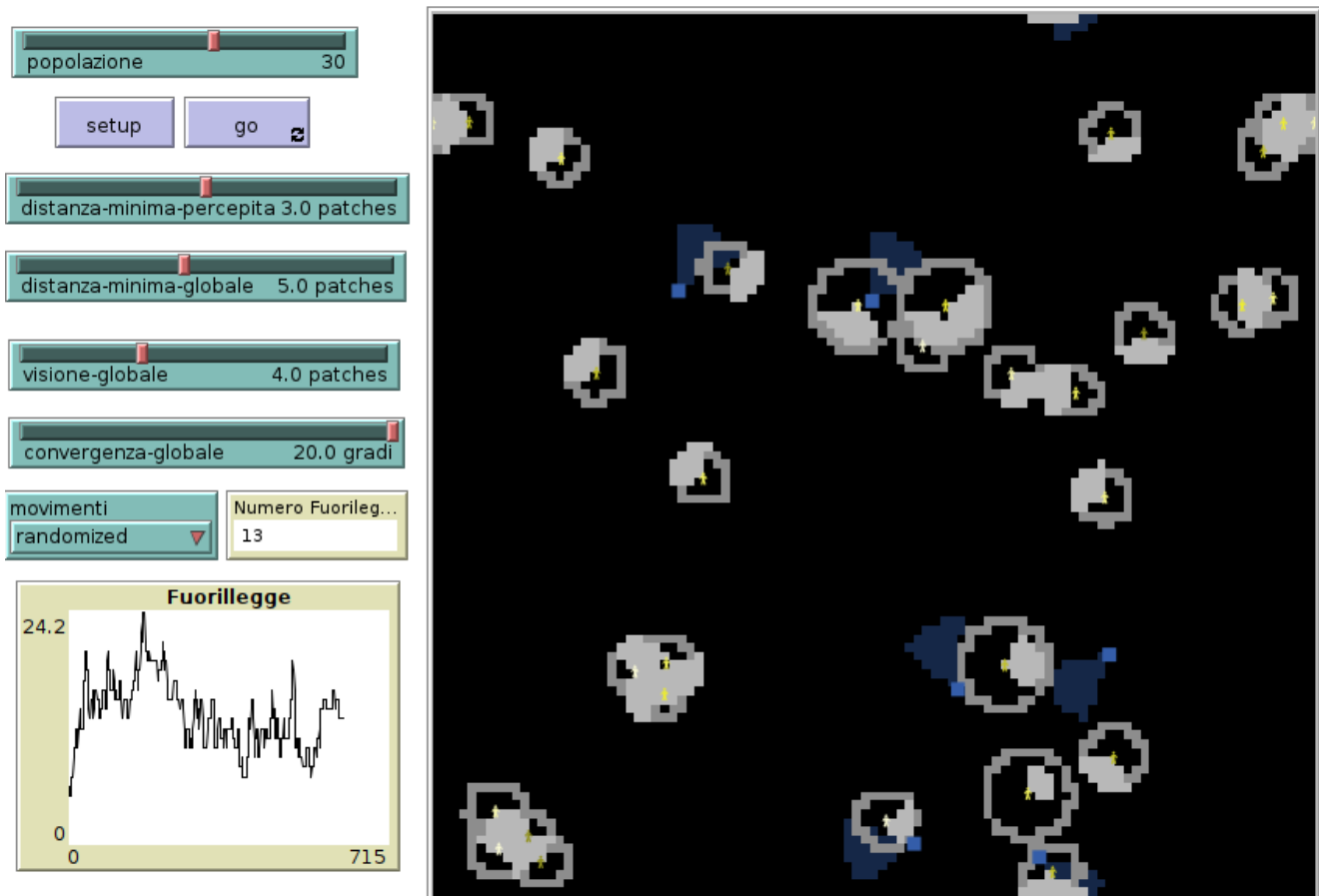
In quest' altra simulazione emerge nella maggior parte dei casi il comportamento opposto, in quanto la lunghezza della visione è minore della distanza minima percepita dalle singole persone. Di conseguenza se le persone non riescono a vedere oltre il raggio della propria distanza minima, non riusciranno mai a formare assembramenti, perchè appena entra almeno una persona nel loro campo visivo si separeranno immediatamente per rispettare le regole invece che fermarsi.



I casi presi in considerazione per studiare il funzionamento della vista sono casi estremi, che mostrano come sia importante questo parametro per gli agenti influenzandone il comportamento anche più della distanza minima di sicurezza. Un buon utilizzo di questo parametro, per renderlo il più veritiero possibile è quello di impostare il valore maggiore della distanza minima percepita, ma minore di quella globale.

### 3.5 Distanza minima percepita minore della globale

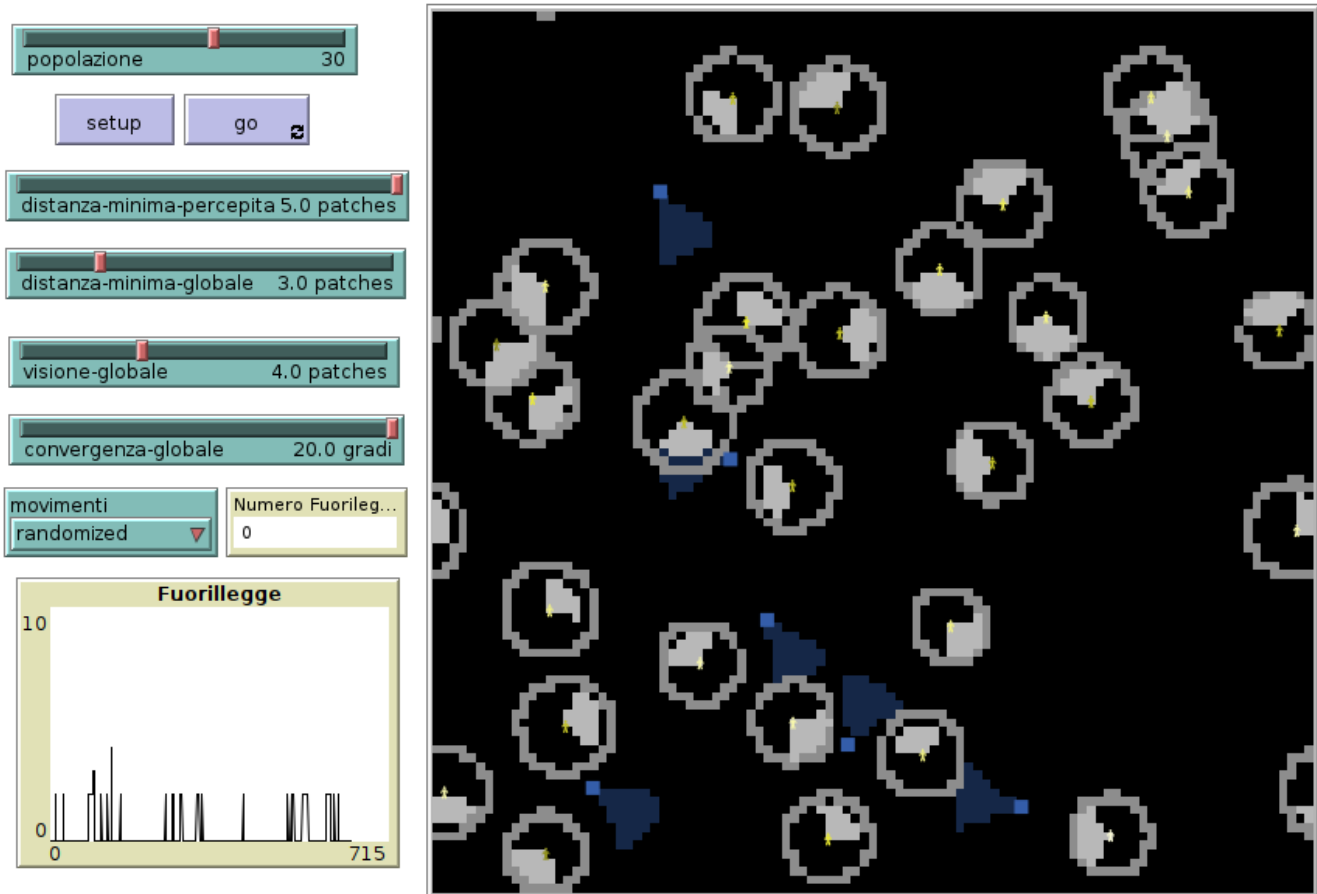
Impostando la vista come appena indicato, utilizzando una distanza minima percepita più bassa rispetto a quella globale, il comportamento delle persone sarà quello di formare degli assembramenti e aumenterà il numero di persone fuorilegge. Di conseguenza, i distanziatori passando attraverso gli assembramenti, tenderanno a distanziare le persone fuorilegge nel loro raggio visivo.



In questo caso evinciamo il comportamento, reale, delle persone quando sottovalutano il problema e vengono scoperti dalle forze dell'ordine.

### 3.6 Distanza minima percepita maggiore della globale

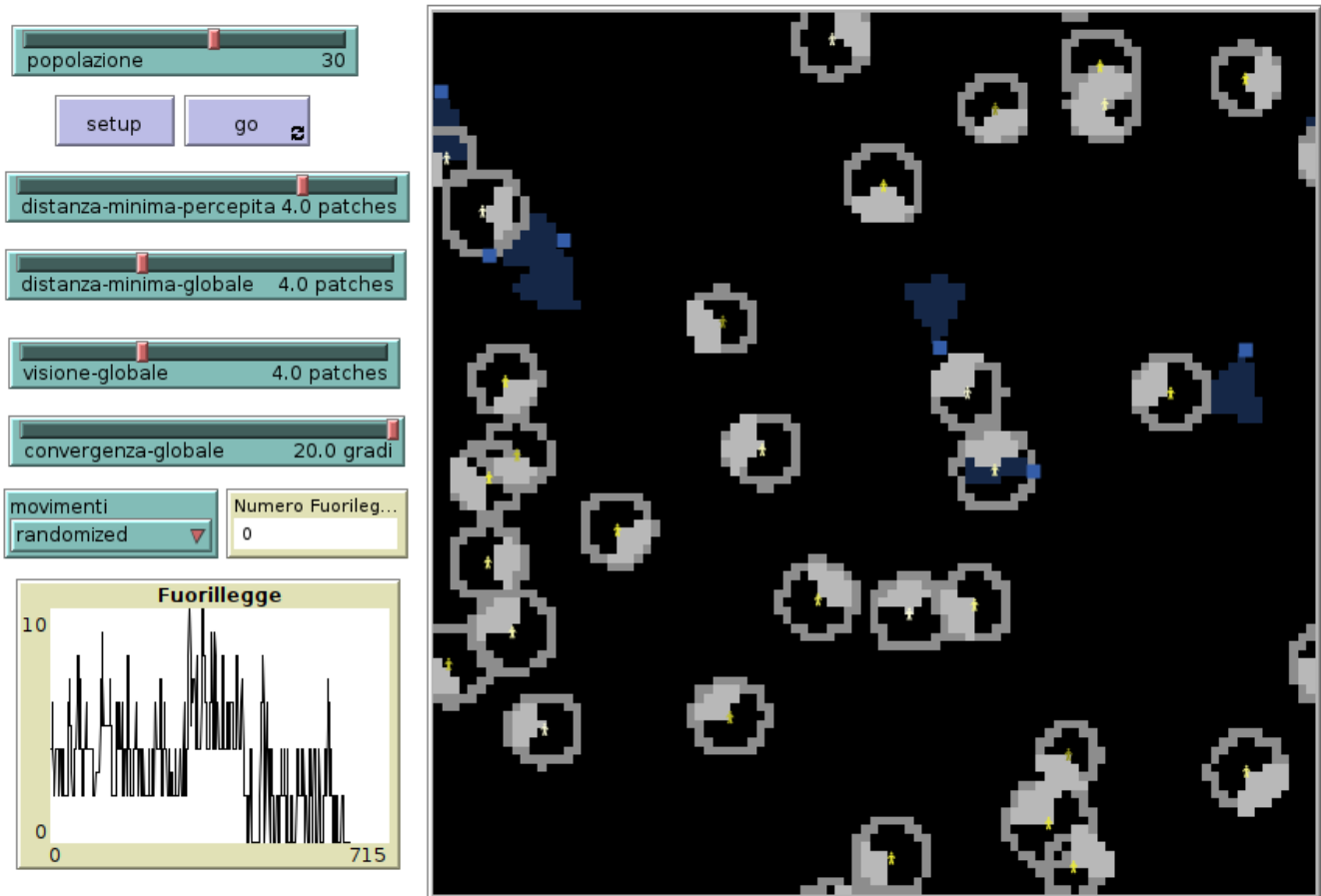
Mantenendo gli stessi valori del test precedente, ma aumentando la distanza minima percepita fino a superare quella globale, le persone non formeranno mai assembramenti nella maggiorparte dei casi.



Questo simula il comportamento delle persone quando sono molto spaventate dal virus e arrivano a sopravvalutare la distanza di sicurezza.

### 3.7 Distanza minima percepita uguale alla globale

In questo test, la distanza minima percepita è stata impostata uguale a quella globale. Quindi non ci sono da parte delle persone, né una sottovalutazione del problema né tantomeno una sopravvalutazione. In questo caso è difficile che avvenga la formazione di assembramenti, se non utilizzando una vista maggiore della distanza in modo da simulare una necessità di altre persone.



Le persone sono consapevoli del rischio che deriva dal non mantenere la distanza quindi se sentissero il bisogno di avvicinarsi alle altre persone lo farebbero mantenendo le distanze in maniera corretta.

# Capitolo 4

## Conclusioni

Le ipotesi fatte nel primo capitolo si sono verificate essere valide e precise.

La distanza minima percepita e quella globale si sono verificate essere un ottimo modo per giocare col comportamento delle persone inquanto hanno influenzato maggiormente la formazione di assembramenti.

Una nota interessante è sorta tramite l'utilizzo combinato della visione e della convergenza offrendo così maggiore impatto sulla tendenza delle persone ad attrarsi.

Il comportamento dei distanziatori, messi a simulare le forze dell'ordine, ha dato maggiori frutti nel momento in cui le persone non avessero modo di posizionarsi in punti in cui non venivano viste. Si potrebbe ampliare, in futuro, la modalità di visione dei distanziatori, in modo da poter convergere verso gli assembramenti.

Il modello sviluppato risulta, comunque, essere un ottimo punto di partenza per effettuare degli studi sugli assembramenti e sul rispettivo modo per evitarli e contrastarli. Abbiamo raggiunto una fedeltà, non perfetta ma comunque ottima in quanto alcune situazioni verificatesi nei test sono in qualche modo ricollegabili per comportamenti alla vita reale.

Per esempio nel test *Distanza minima percepita uguale o simile alla globale* abbiamo riscontrato quanto successo in Cina, in cui il rispetto delle regole era molto rigido e le persone avevano una bassa tendenza a convergere verso le altre.

Un altro esempio è quello del test *Distanza minima percepita minore della globale* dove abbiamo riscontrato con facilità il verificarsi di assembramenti. Questo caso è paragonabile ai centri storici durante le sere degli weekend in cui le persone giovani tendevano ad ammassarsi senza rispettare le distanze di sicurezza.