**Model-1**

globals

[

maximum-infectious ;; The maximum number of infectious individuals at one simulation tick.

tick-at-maximum-infectious ;; The (first) tick when the maximum number of infectious individuals is realized.

number-infectious-vector ;; Vector of the number of infectious individuals at each simulation tick.

incubation-alpha ;; Alpha parameter for the gamma distribution used in calculating incubation-time.

incubation-lambda ;; Lambda parameter for the gamma distribution used in calculating incubation-time.

infectious-alpha ;; Alpha parameter for the gamma distribution used in calculating infectious-time.

infectious-lambda ;; Lambda parameter for the gamma distribution used in calculating infectious-time.

number-of-recovery

number-of-death

number-of-quarantine

]

turtles-own

[

age

susceptible? ;; If true, the individual is a member of the susceptible class.

exposed? ;; If true, the individual is a member of the exposed (incubation) class.

infectious? ;; If true, the individual is a member of the infectious class.

recovered? ;; If true, the individual is a member of the recovered class.

isolation? ;; If true, the individual is in isolation

close-contact?

incubation-length ;; How long the individual has been in the exposed class, increasing by 1 each tick. This is compared against the incubation-time, selected from a gamma-distribution.

incubation-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the exposed class.

infectious-length ;; How long the individual has been in the infectious class, increasing by 1 each tick. This is compared against the infectious-time, selected from a gamma-distribution.

infectious-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the infectious class.

isolation-length ;; How long the close contact has been isolated

isolation-time ;; it usually fixed at 14 days

isolation-delay-time

quarantine-delay-time

total-contacts ;; A count of all contacts of the individual.

total-not-isolated-contacts

]

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Setup Procedures ;;;;

to setup

clear-all

setup-gamma-distributions

setup-population

setup-number

reset-ticks

end

to setup-gamma-distributions ;; The calculation from mean and standard deviation (in days) to the alpha and lambda parameters required for the gamma-distributions (in ticks).

set incubation-alpha (average-incubation-period \* ticks-per-day)^ 2 / (incubation-standard-deviation \* ticks-per-day)^ 2

set incubation-lambda (average-incubation-period \* ticks-per-day) / (incubation-standard-deviation \* ticks-per-day)^ 2

set infectious-alpha (average-infectious-period \* ticks-per-day)^ 2 / (infectious-standard-deviation \* ticks-per-day)^ 2

set infectious-lambda (average-infectious-period \* ticks-per-day) / (infectious-standard-deviation \* ticks-per-day)^ 2

end

to setup-population

create-turtles initial-population

[

setxy random-xcor random-ycor ;; All individuals are placed on random patches in the world.

set susceptible? true ;; All individuals are set as susceptible.

set exposed? false

set infectious? false

set recovered? false

set isolation? false

set close-contact? false

set shape "person"

set total-contacts 0

set total-not-isolated-contacts 0

ask turtle 0 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 150 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 200 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 250 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

set number-infectious-vector [ 1 ] ;; The number-infectious-vector vector is initiallized.

assign-color

]

end

to assign-color

if susceptible?

[ set color white ]

if exposed?

[ set color yellow ]

if infectious?

[ set color red ]

if recovered?

[ set color lime ]

end

to setup-number

set number-of-recovery 0

set number-of-death 0

set number-of-quarantine 0

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Go Procedure ;;;;

to go

if (all? turtles [susceptible? and not close-contact? or recovered?] ) ;; The simulation ends when no individuals are infected (exposed or infectious).

[ stop ]

ask turtles with [not isolation?]

[ move]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might expose susceptible neighbors.

[ expose-neighbors ]

ask turtles with [ infectious? ] ;; If infectious individuals have been infectious for infectious-time ticks, they will recover.

[

chance-of-recovery ]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might isolate.

[ set-close-contact]

ask turtles with [ infectious? and not isolation? ] with [not close-contact?] ;; Infectious and non-isolated individuals might isolate.

[ chance-of-isolating]

ask turtles with [ isolation? and not infectious? ] ;; Individuals in isolation that recover, exit isolation.

[ exit-isolation ]

ask turtles with [ exposed? and not close-contact?] ;; If exposed individuals have been in the exposed class for incubation-time ticks, they will become infectious.

[ chance-of-becoming-infectious ]

quarantine-close-contact

close-contact-exposed

quarantine-exposed

quarantine-infectious

ask turtles with [close-contact?] with [not exposed? or not infectious?] with [isolation?]

[exit-isolation-close-contact]

ask turtles

[ assign-color

count-contacts ]

compute-maximum-infectious

compute-quarantine

tick

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Nested Functions ;;;;

to move ;; Individuals turn a random angle between -40 and 40 degrees then step forward 1 unit.

right (random 80) - 40

forward 1

if not can-move? 1 [ right 180 ] ;; If an individual is at the world's boundary, it turns around.

end

to count-contacts ;; Contacts are defined as other individuals within a 1 unit radius. Not-isolated-contacts are between two individuals both not in isolation.

set total-contacts total-contacts + count other turtles with [distance myself <= infect-radius]

if not isolation?

[

set total-not-isolated-contacts total-not-isolated-contacts + count other turtles with [distance myself <= infect-radius] with [ not isolation? ]

]

end

to expose-neighbors

ask turtles with [infectious? and not isolation?]

[

let infecter self

let nearby-uninfected turtles with [susceptible?] with [distance myself <= infect-radius]

if (nearby-uninfected != nobody) [

ask nearby-uninfected [

if (random-float 1 < transmission-constant \* (( 1 - distance myself / infect-radius)^ Exponent-transmission-rates))

[ if random-float 100 > handwashing-and-mask

[set susceptible? false

set exposed? true

set incubation-time random-gamma incubation-alpha incubation-lambda ;; A newly exposed individual selects an incubation-time from the gamma-distribution and its incubation-length is set to 0.

set incubation-length 0

]

]

]

]

]

end

to chance-of-becoming-infectious ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to chance-of-recovery ;; When an infectious individual has been in the infectious class longer than its infection-time, it will recover.

set infectious-length infectious-length + 1

if infectious-length > infectious-time

[

set number-of-recovery number-of-recovery + 1

set infectious? false

set recovered? true

]

end

to chance-of-isolating ;; Once each day, an infectious individual will isolate with probability isolation-chance.

if ticks / ticks-per-day mod 1 = 0

[ ifelse isolation-delay-time <= 0

[

set isolation? true

set pcolor gray

]

[set isolation-delay-time isolation-delay-time - 1]

]

end

to exit-isolation ;; After an isolated individual recovers, they leave the isolation class.

if recovered?

[

set isolation? false

set pcolor black

]

end

to set-close-contact

ask turtles with [infectious? and not isolation?]

[

let infecter self

let close-contact turtles with [distance myself <= infect-radius] with [not infectious? and not isolation?]

if (close-contact != nobody) [

ask close-contact

[set close-contact? true

set quarantine-delay-time average-isolation-delay-time

]

]

]

end

to quarantine-close-contact

ask turtles with [close-contact? and not isolation?] with [susceptible?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[

set susceptible? false

set isolation? true

set pcolor orange

set isolation-time 0

set isolation-length 14 \* ticks-per-day]

]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to close-contact-exposed

ask turtles with [close-contact? ] with [exposed?]

[chance-of-becoming-infectious-exposed]

end

to quarantine-exposed

ask turtles with [close-contact? and not isolation?] with [exposed?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to chance-of-becoming-infectious-exposed ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to quarantine-infectious

ask turtles with [close-contact? and not isolation?] with [infectious?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to exit-isolation-close-contact

ifelse isolation-time >= isolation-length

[set isolation? false

set close-contact? false

set susceptible? true

set pcolor black]

[set isolation-time isolation-time + 1]

end

to compute-maximum-infectious ;; A vector of the number of infectious individuals at each tick is stored. The maximum and time of the maximum are computed.

set number-infectious-vector lput count turtles with [infectious?] number-infectious-vector

set maximum-infectious max number-infectious-vector

set tick-at-maximum-infectious position maximum-infectious number-infectious-vector

end

to compute-quarantine

set number-of-quarantine number-of-quarantine + count turtles with [isolation? and not infectious?]

end

**Model-2**

globals

[

maximum-infectious ;; The maximum number of infectious individuals at one simulation tick.

tick-at-maximum-infectious ;; The (first) tick when the maximum number of infectious individuals is realized.

number-infectious-vector ;; Vector of the number of infectious individuals at each simulation tick.

incubation-alpha ;; Alpha parameter for the gamma distribution used in calculating incubation-time.

incubation-lambda ;; Lambda parameter for the gamma distribution used in calculating incubation-time.

infectious-alpha ;; Alpha parameter for the gamma distribution used in calculating infectious-time.

infectious-lambda ;; Lambda parameter for the gamma distribution used in calculating infectious-time.

number-of-recovery

number-of-death

number-of-quarantine

]

turtles-own

[

age

susceptible? ;; If true, the individual is a member of the susceptible class.

exposed? ;; If true, the individual is a member of the exposed (incubation) class.

infectious? ;; If true, the individual is a member of the infectious class.

recovered? ;; If true, the individual is a member of the recovered class.

isolation? ;; If true, the individual is in isolation

close-contact?

incubation-length ;; How long the individual has been in the exposed class, increasing by 1 each tick. This is compared against the incubation-time, selected from a gamma-distribution.

incubation-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the exposed class.

infectious-length ;; How long the individual has been in the infectious class, increasing by 1 each tick. This is compared against the infectious-time, selected from a gamma-distribution.

infectious-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the infectious class.

isolation-length ;; How long the close contact has been isolated

isolation-time ;; it usually fixed at 14 days

isolation-delay-time

quarantine-delay-time

total-contacts ;; A count of all contacts of the individual.

total-not-isolated-contacts

]

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Setup Procedures ;;;;

to setup

clear-all

setup-gamma-distributions

setup-population

setup-number

reset-ticks

end

to setup-gamma-distributions ;; The calculation from mean and standard deviation (in days) to the alpha and lambda parameters required for the gamma-distributions (in ticks).

set incubation-alpha (average-incubation-period \* ticks-per-day)^ 2 / (incubation-standard-deviation \* ticks-per-day)^ 2

set incubation-lambda (average-incubation-period \* ticks-per-day) / (incubation-standard-deviation \* ticks-per-day)^ 2

set infectious-alpha (average-infectious-period \* ticks-per-day)^ 2 / (infectious-standard-deviation \* ticks-per-day)^ 2

set infectious-lambda (average-infectious-period \* ticks-per-day) / (infectious-standard-deviation \* ticks-per-day)^ 2

end

to setup-population

create-turtles initial-population

[

setxy random-xcor random-ycor ;; All individuals are placed on random patches in the world.

set susceptible? true ;; All individuals are set as susceptible.

set exposed? false

set infectious? false

set recovered? false

set isolation? false

set close-contact? false

set shape "person"

set total-contacts 0

set total-not-isolated-contacts 0

ask turtle 0 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 150 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 200 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 250 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

set number-infectious-vector [ 1 ] ;; The number-infectious-vector vector is initiallized.

assign-color

]

end

to assign-color

if susceptible?

[ set color white ]

if exposed?

[ set color yellow ]

if infectious?

[ set color red ]

if recovered?

[ set color lime ]

end

to setup-number

set number-of-recovery 0

set number-of-death 0

set number-of-quarantine 0

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Go Procedure ;;;;

to go

if (all? turtles [susceptible? and not close-contact? or recovered?] ) ;; The simulation ends when no individuals are infected (exposed or infectious).

[ stop ]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might expose susceptible neighbors.

[ expose-neighbors ]

ask turtles with [ infectious? ] ;; If infectious individuals have been infectious for infectious-time ticks, they will recover.

[

chance-of-recovery ]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might isolate.

[ set-close-contact]

ask turtles with [ infectious? and not isolation? ] with [not close-contact?] ;; Infectious and non-isolated individuals might isolate.

[ chance-of-isolating]

ask turtles with [ isolation? and not infectious? ] ;; Individuals in isolation that recover, exit isolation.

[ exit-isolation ]

ask turtles with [ exposed? and not close-contact?] ;; If exposed individuals have been in the exposed class for incubation-time ticks, they will become infectious.

[ chance-of-becoming-infectious ]

ask turtles with [not isolation?]

[ gathering-restriction]

quarantine-close-contact

close-contact-exposed

quarantine-exposed

quarantine-infectious

ask turtles with [close-contact?] with [not exposed? or not infectious?] with [isolation?]

[exit-isolation-close-contact]

ask turtles

[ assign-color

count-contacts ]

compute-maximum-infectious

compute-quarantine

tick

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Nested Functions ;;;;

to gathering-restriction

let others self

let nearby-agent turtles with [distance myself <= infect-radius]

if (nearby-agent != nobody) [

ask nearby-agent [

right 90

forward 1

if not can-move? 1 [ right 180 ] ;; If an individual is at the world's boundary, it turns around.

]

]

end

to count-contacts ;; Contacts are defined as other individuals within a 1 unit radius. Not-isolated-contacts are between two individuals both not in isolation.

set total-contacts total-contacts + count other turtles with [distance myself <= infect-radius]

if not isolation?

[

set total-not-isolated-contacts total-not-isolated-contacts + count other turtles with [distance myself <= infect-radius] with [ not isolation? ]

]

end

to expose-neighbors

ask turtles with [infectious? and not isolation?]

[

let infecter self

let nearby-uninfected turtles with [susceptible?] with [distance myself <= infect-radius]

if (nearby-uninfected != nobody) [

ask nearby-uninfected [

if (random-float 1 < transmission-constant \* (( 1 - distance myself / infect-radius)^ Exponent-transmission-rates))

[ if random-float 100 > handwashing-and-mask

[set susceptible? false

set exposed? true

set incubation-time random-gamma incubation-alpha incubation-lambda ;; A newly exposed individual selects an incubation-time from the gamma-distribution and its incubation-length is set to 0.

set incubation-length 0

]

]

]

]

]

end

to chance-of-becoming-infectious ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to chance-of-recovery ;; When an infectious individual has been in the infectious class longer than its infection-time, it will recover.

set infectious-length infectious-length + 1

if infectious-length > infectious-time

[

set number-of-recovery number-of-recovery + 1

set infectious? false

set recovered? true

]

end

to chance-of-isolating ;; Once each day, an infectious individual will isolate with probability isolation-chance.

if ticks / ticks-per-day mod 1 = 0

[ ifelse isolation-delay-time <= 0

[

set isolation? true

set pcolor gray

]

[set isolation-delay-time isolation-delay-time - 1]

]

end

to exit-isolation ;; After an isolated individual recovers, they leave the isolation class.

if recovered?

[

set isolation? false

set pcolor black

]

end

to set-close-contact

ask turtles with [infectious? and not isolation?]

[

let infecter self

let close-contact turtles with [distance myself <= infect-radius] with [not infectious? and not isolation?]

if (close-contact != nobody) [

ask close-contact

[set close-contact? true

set quarantine-delay-time average-isolation-delay-time

]

]

]

end

to quarantine-close-contact

ask turtles with [close-contact? and not isolation?] with [susceptible?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[

set susceptible? false

set isolation? true

set pcolor orange

set isolation-time 0

set isolation-length 14 \* ticks-per-day]

]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to close-contact-exposed

ask turtles with [close-contact? ] with [exposed?]

[chance-of-becoming-infectious-exposed]

end

to quarantine-exposed

ask turtles with [close-contact? and not isolation?] with [exposed?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to chance-of-becoming-infectious-exposed ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to quarantine-infectious

ask turtles with [close-contact? and not isolation?] with [infectious?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to exit-isolation-close-contact

ifelse isolation-time >= isolation-length

[set isolation? false

set close-contact? false

set susceptible? true

set pcolor black]

[set isolation-time isolation-time + 1]

end

to compute-maximum-infectious ;; A vector of the number of infectious individuals at each tick is stored. The maximum and time of the maximum are computed.

set number-infectious-vector lput count turtles with [infectious?] number-infectious-vector

set maximum-infectious max number-infectious-vector

set tick-at-maximum-infectious position maximum-infectious number-infectious-vector

end

to compute-quarantine

set number-of-quarantine number-of-quarantine + count turtles with [isolation? and not infectious?]

end

**Model-3**

globals

[

maximum-infectious ;; The maximum number of infectious individuals at one simulation tick.

tick-at-maximum-infectious ;; The (first) tick when the maximum number of infectious individuals is realized.

number-infectious-vector ;; Vector of the number of infectious individuals at each simulation tick.

incubation-alpha ;; Alpha parameter for the gamma distribution used in calculating incubation-time.

incubation-lambda ;; Lambda parameter for the gamma distribution used in calculating incubation-time.

infectious-alpha ;; Alpha parameter for the gamma distribution used in calculating infectious-time.

infectious-lambda ;; Lambda parameter for the gamma distribution used in calculating infectious-time.

number-of-recovery

number-of-death

number-of-quarantine

]

turtles-own

[

age

susceptible? ;; If true, the individual is a member of the susceptible class.

exposed? ;; If true, the individual is a member of the exposed (incubation) class.

infectious? ;; If true, the individual is a member of the infectious class.

recovered? ;; If true, the individual is a member of the recovered class.

isolation? ;; If true, the individual is in isolation

close-contact?

incubation-length ;; How long the individual has been in the exposed class, increasing by 1 each tick. This is compared against the incubation-time, selected from a gamma-distribution.

incubation-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the exposed class.

infectious-length ;; How long the individual has been in the infectious class, increasing by 1 each tick. This is compared against the infectious-time, selected from a gamma-distribution.

infectious-time ;; The randomly chosen gamma-distribution value for how long the individual will be in the infectious class.

isolation-length ;; How long the close contact has been isolated

isolation-time ;; it usually fixed at 14 days

isolation-delay-time

quarantine-delay-time

total-contacts ;; A count of all contacts of the individual.

total-not-isolated-contacts

]

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Setup Procedures ;;;;

to setup

clear-all

setup-gamma-distributions

setup-population

setup-number

reset-ticks

end

to setup-gamma-distributions ;; The calculation from mean and standard deviation (in days) to the alpha and lambda parameters required for the gamma-distributions (in ticks).

set incubation-alpha (average-incubation-period \* ticks-per-day)^ 2 / (incubation-standard-deviation \* ticks-per-day)^ 2

set incubation-lambda (average-incubation-period \* ticks-per-day) / (incubation-standard-deviation \* ticks-per-day)^ 2

set infectious-alpha (average-infectious-period \* ticks-per-day)^ 2 / (infectious-standard-deviation \* ticks-per-day)^ 2

set infectious-lambda (average-infectious-period \* ticks-per-day) / (infectious-standard-deviation \* ticks-per-day)^ 2

end

to setup-population

create-turtles initial-population

[

setxy random-xcor random-ycor ;; All individuals are placed on random patches in the world.

set susceptible? true ;; All individuals are set as susceptible.

set exposed? false

set infectious? false

set recovered? false

set isolation? false

set close-contact? false

set shape "person"

set total-contacts 0

set total-not-isolated-contacts 0

ask turtle 0 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 150 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 200 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

ask turtle 250 ;; Individual 0 begins as infectious. Its infectious-time is selected from the gamma distribution and infectious-length set to 0.

[

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda

set infectious-length 0

]

set number-infectious-vector [ 1 ] ;; The number-infectious-vector vector is initiallized.

assign-color

]

end

to assign-color

if susceptible?

[ set color white ]

if exposed?

[ set color yellow ]

if infectious?

[ set color red ]

if recovered?

[ set color lime ]

end

to setup-number

set number-of-recovery 0

set number-of-death 0

set number-of-quarantine 0

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Go Procedure ;;;;

to go

if (all? turtles [susceptible? and not close-contact? or recovered?] ) ;; The simulation ends when no individuals are infected (exposed or infectious).

[ stop ]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might expose susceptible neighbors.

[ expose-neighbors ]

ask turtles with [ infectious? ] ;; If infectious individuals have been infectious for infectious-time ticks, they will recover.

[

chance-of-recovery ]

ask turtles with [ infectious? and not isolation? ] ;; Infectious and non-isolated individuals might isolate.

[ set-close-contact]

ask turtles with [ infectious? and not isolation? ] with [not close-contact?] ;; Infectious and non-isolated individuals might isolate.

[ chance-of-isolating]

ask turtles with [ isolation? and not infectious? ] ;; Individuals in isolation that recover, exit isolation.

[ exit-isolation ]

ask turtles with [ exposed? and not close-contact?] ;; If exposed individuals have been in the exposed class for incubation-time ticks, they will become infectious.

[ chance-of-becoming-infectious ]

quarantine-close-contact

close-contact-exposed

quarantine-exposed

quarantine-infectious

ask turtles with [close-contact?] with [not exposed? or not infectious?] with [isolation?]

[exit-isolation-close-contact]

ask turtles

[ assign-color

count-contacts ]

compute-maximum-infectious

compute-quarantine

tick

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;; Nested Functions ;;;;

to count-contacts ;; Contacts are defined as other individuals within a 1 unit radius. Not-isolated-contacts are between two individuals both not in isolation.

set total-contacts total-contacts + count other turtles with [distance myself <= infect-radius]

if not isolation?

[

set total-not-isolated-contacts total-not-isolated-contacts + count other turtles with [distance myself <= infect-radius] with [ not isolation? ]

]

end

to expose-neighbors

ask turtles with [infectious? and not isolation?]

[

let infecter self

let nearby-uninfected turtles with [susceptible?] with [distance myself <= infect-radius]

if (nearby-uninfected != nobody) [

ask nearby-uninfected [

if (random-float 1 < transmission-constant \* (( 1 - distance myself / infect-radius)^ Exponent-transmission-rates))

[ if random-float 100 > handwashing-and-mask

[set susceptible? false

set exposed? true

set incubation-time random-gamma incubation-alpha incubation-lambda ;; A newly exposed individual selects an incubation-time from the gamma-distribution and its incubation-length is set to 0.

set incubation-length 0

]

]

]

]

]

end

to chance-of-becoming-infectious ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to chance-of-recovery ;; When an infectious individual has been in the infectious class longer than its infection-time, it will recover.

set infectious-length infectious-length + 1

if infectious-length > infectious-time

[

set number-of-recovery number-of-recovery + 1

set infectious? false

set recovered? true

]

end

to chance-of-isolating ;; Once each day, an infectious individual will isolate with probability isolation-chance.

if ticks / ticks-per-day mod 1 = 0

[ ifelse isolation-delay-time <= 0

[

set isolation? true

set pcolor gray

]

[set isolation-delay-time isolation-delay-time - 1]

]

end

to exit-isolation ;; After an isolated individual recovers, they leave the isolation class.

if recovered?

[

set isolation? false

set pcolor black

]

end

to set-close-contact

ask turtles with [infectious? and not isolation?]

[

let infecter self

let close-contact turtles with [distance myself <= infect-radius] with [not infectious? and not isolation?]

if (close-contact != nobody) [

ask close-contact

[set close-contact? true

set quarantine-delay-time average-isolation-delay-time

]

]

]

end

to quarantine-close-contact

ask turtles with [close-contact? and not isolation?] with [susceptible?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[

set susceptible? false

set isolation? true

set pcolor orange

set isolation-time 0

set isolation-length 14 \* ticks-per-day]

]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to close-contact-exposed

ask turtles with [close-contact? ] with [exposed?]

[chance-of-becoming-infectious-exposed]

end

to quarantine-exposed

ask turtles with [close-contact? and not isolation?] with [exposed?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to chance-of-becoming-infectious-exposed ;; When an infected individual has been in the exposed class longer than its incubation-time, it will become infectious.

set incubation-length incubation-length + 1

if incubation-length > incubation-time

[

set exposed? false

set susceptible? false

set infectious? true

set infectious-time random-gamma infectious-alpha infectious-lambda ;; A newly infectious individual selects an infectious-time from the gamma-distribution and its infection-length is set to 0.

set infectious-length 0

]

end

to quarantine-infectious

ask turtles with [close-contact? and not isolation?] with [infectious?]

[ifelse quarantine-delay-time <= 0

[if (random-float 100 < isolation-and-quarantine-chance)

[ set isolation? true

set pcolor orange

set close-contact? false

]]

[set quarantine-delay-time quarantine-delay-time - 1]

]

end

to exit-isolation-close-contact

ifelse isolation-time >= isolation-length

[set isolation? false

set close-contact? false

set susceptible? true

set pcolor black]

[set isolation-time isolation-time + 1]

end

to compute-maximum-infectious ;; A vector of the number of infectious individuals at each tick is stored. The maximum and time of the maximum are computed.

set number-infectious-vector lput count turtles with [infectious?] number-infectious-vector

set maximum-infectious max number-infectious-vector

set tick-at-maximum-infectious position maximum-infectious number-infectious-vector

end

to compute-quarantine

set number-of-quarantine number-of-quarantine + count turtles with [isolation? and not infectious?]

end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;