

```

turtles-own[
  insured?
  checked?
  numberofedges
  payoff
]
globals[
  numberofinsued
  numberofnotinsured
  donewithinsured?
  donewithnotinsured?
]
to setup
  clear-all
  setup-turtles
  reset-ticks
  set numberofinsued 0
  set donewithinsured? false
  set donewithnotinsured? false
  setup-patches
end

to setup-turtles
  set-default-shape turtles "circle"

  crt num-nodes
  layout-circle turtles max-pxcor - 20
  ask turtles [
    set payoff 0
    set insured? false
    set checked? false
    set color red

    if (random-float 100.0 < (prob-
insured)) [
      set color green
      set insured? true
      set numberofinsued (numberofinsued
+ 1)
    ]
  ]
  ;ask turtles [ set label who set label-
color black]

```

```

end

to setup-patches
  ask patches [
    set pcolor white
  ]
end

to go
  if not donewithinsured? [
    add-edge
  ]
  tick
end

to add-edge
  let node1 one-of turtles with [not
checked?]
  if node1 = nobody
  [
    display
    user-message "insured clique finished"
    stop
  ]
  ask node1 [
    let node2 one-of turtles with [not link-
neighbor? node1 and (self != node1) and
not checked?]

    ifelse node2 = nobody
    [
      set checked? true
      add-edge
    ]
    [
      let nolinkpayoff payoff
      ifelse insured?
      [
        ;node1 is insured
        ask node2
        [
          let nolinkpayoff2 payoff
          ifelse insured?
          [

```

```

;node2 and node1 insured
let newpayoff1 (nolinkpayoff +
(beta / 100) - (insurancelink / 100))
let newpayoff2 (nolinkpayoff2 +
(beta / 100) - (insurancelink / 100))
if newpayoff1 > nolinkpayoff and
newpayoff2 > nolinkpayoff2
[
;add link
create-link-with node1
set payoff newpayoff2
ask node1[
set payoff newpayoff1

]
]
;done with adding link
]
[;begin else
;node2 not insured
let newpayoff1 (nolinkpayoff +
(beta / 100) - (risk / 100) - (insurancelink
/ 100))
let newpayoff2 (nolinkpayoff2 +
(beta / 100))
if newpayoff1 > nolinkpayoff and
newpayoff2 > nolinkpayoff2
[
;add link
create-link-with node1
set payoff newpayoff2
ask node1[
set payoff newpayoff1
]
]
;done with adding link

];end else
];done with node2
]
[
;node1 not insured
ask node2
[

```

```

let nolinkpayoff2 payoff
ifelse insured?
[
;node2 insured and node1 not
insured
let newpayoff1 (nolinkpayoff +
(beta / 100))
let newpayoff2 (nolinkpayoff2 +
(beta / 100) - (risk / 100) - (insurancelink
/ 100))
if newpayoff1 > nolinkpayoff and
newpayoff2 > nolinkpayoff2
[
;add link
create-link-with node1
set payoff newpayoff2
ask node1[
set payoff newpayoff1

]
]
;done with adding link
]
[;begin else
;node2 and node1 not insured
let newpayoff1 (nolinkpayoff +
(beta / 100) - (risk / 100))
let newpayoff2 (nolinkpayoff2 +
(beta / 100) - (risk / 100))
if newpayoff1 > nolinkpayoff and
newpayoff2 > nolinkpayoff2
[
;add link
create-link-with node1
set payoff newpayoff2
ask node1[
set payoff newpayoff1
]
]
;done with adding link

];end else
];done with node2

```

```

    ]
    ;set color green
    ;add-edge
  ]
]
layout
end

```

```

to add-edge-not-insured
  let node1 one-of turtles with [not
insured? and not checked?]
  if node1 = nobody
  [
    ;display
    ;user-message "non-insured clique
finished"
    stop
  ]
  ask node1[
    let node2 one-of turtles with [not
insured? and not link-neighbor? node1
and (self != node1) and not checked?]
    ifelse node2 = nobody
    [
      display
      set donewithnotinsured? true
      set checked? true
      add-edge-not-insured
    ]
    [
      create-link-with node2
      add-edge-not-insured
    ]
  ]
  layout
end

```

```

to layout
  repeat 10 [
    layout-spring (turtles with [any? link-
neighbors]) links 0.4 6 1
    display ;; so we get smooth animation
  ]
end

```