```
globals [
  source-intensity
  max-pheromone-attract
  max-pheromone-repel
  max-chemical
  to-RGB ; for representing multiple chemical gradients on a single patch
  step-size
  threshold-to-communicate
]
patches-own [
  intensity
  intensity-change
  source?
  wall?
  pheromone-attract
  pheromone-change
  pheromone-repel
]
turtles-own [
  velocity
  time-to-source
  time-in-radius

  ; relating to the bacteria's
  ; detection of the food source gradient
  previous-intensity
  source-gradient
  source-change-perceived

  ; and detection of the pheromone gradient
  ; (communication between bacteria cells)
  previous-pheromone
  pheromone-gradient
  pheromone-change-perceived

]


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;; Setup procedures ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup
  clear-all
  set source-intensity 10

  setup-patches
  setup-turtles
  color-patches
  reset-ticks
end



to setup-patches
  ask patches [
    set intensity 0
    set intensity-change 0
    set source? false
    set wall? false

    set pheromone-attract 0
    set pheromone-change 0
    set pheromone-repel 0
    set to-RGB 255 / white

    set max-chemical 5
    set max-pheromone-attract 10
    set max-pheromone-repel 10
  ]

  if obstacles = true [
    ask patches [
      let coin random 40
```

```
      set pcolor black
      if (coin = 0) [
        set wall? true
        set pcolor white
        ask neighbors4 [
          set wall? true
          set pcolor white
        ]
      ]
    ]
  ]
  setup-sources

end

to setup-sources
  ask patch 50 50 [
    set source? True
    set wall? false
    set intensity source-intensity
    set pcolor yellow
  ]
end

to setup-turtles
  set-default-shape turtles "circle"
  create-turtles number-of-agents [
    set heading (random 360)
    setxy random-xcor random-ycor
    set time-to-source 99999
    set time-in-radius []
    set threshold-to-communicate 0.01
    ifelse draw-paths [pen-down] [pen-up]
  ]
end

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;; Go procedures ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go
  if ticks > 10000 [stop]

  simulate-chemical
  simulate-pheromone
  simulate-turtles
  color-patches
  tick
end

; ~~~~~~~~~~~~~~~ turtle go procedures ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to simulate-turtles
  ask turtles [

    set previous-intensity intensity
    set previous-pheromone pheromone-attract


    if patch-ahead 1 != nobody and [wall?] of patch-ahead 1 = false [move]

    detect-source-gradient ; bacteria's own detection of the source gradient
    detect-pheromone-gradient ; attractive pheromone gradient change detected by the bacteria (communication


    change-angle

    ; if a positive food source gradient is detected, release an attraction pheromone
    if (source-change-perceived >= threshold-to-communicate )
      [secrete-pheromone]




    ; if the bacteria reaches the source, record the time, first occurrence
    if time-to-source = 99999  and [source? = True] of patch-here  [
```

```
      set time-to-source ticks
    ]
    ; Since we only record the time the bacteria takes to get to the source the first time, I thought
    ; we might want a way to quantify how long the bacteria actually stays by the source.
    if [source? = True] of patch-here [
      let patches-in-radius patches in-radius radius
      if member? patch-here patches-in-radius [ set time-in-radius lput ticks time-in-radius ]
    ]
  ]
end


; The bacteria's "run":
to move
  set step-size 1
  forward step-size
end


; The bacteria's "tumble":
; The angle change of the bacteria takes into account 1) the food source gradient it detects
; 2) the pheromone gradient (from the other bacteria) it detects.
to change-angle
  let previous-angle heading

  ; weighted sum between pheromone and source gradient
  let sumchange max list 0 (source-change-perceived + pheromone-change-perceived) ; want limit to be bigger t
  let std 360 * e ^ (-1 * chemical-sensitivity-of-agents * sumchange)

  set heading random-normal previous-angle std
  ; have the angle depend continuously on the sum of the changes in the gradients
end



; Changes in perception related to changes in physical stimulus can be represented by the Weber-Fechner Law [
;  which states, that the perceived changes in odor concentration are proportional to the log of stimulus
; increase. Therefore, a proxy for the signal the bacteria extracts from the enviornment is: (1/C)*(change_ir
; where here, C is the concentration of the chemical of interest (be it pheromone or food source).
; Go commands "detect-pheromone-gradient" and "detect-source-gradient" are written using this logic.

to detect-source-gradient
  set source-gradient (intensity - previous-intensity)
  set source-change-perceived ((1 / (intensity + 1) ) * (source-gradient)) ; (C + 1) to avoid a 1/0 error.
end

to detect-pheromone-gradient
  set pheromone-gradient (pheromone-attract - previous-pheromone)
  set pheromone-change-perceived (( 1 / (pheromone-attract + 1)) * (pheromone-gradient)) ; (C + 1) to avoid a
end

; secretes the pheromone used to communicate to the other bacteria that it has discovered a positive food sou
to secrete-pheromone
  if communication = true [
    set pheromone-attract pheromone-attract + 5
  ]
end


; ~~~~~~~~~~~~~~~~ patch go procedures ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; Simulate the physics of the chemicals
to simulate-chemical
  diffuse-chemical
  ask patches [
    set intensity (intensity * (100 - 0.1) / 100) ; evaporation
  ]
  ask patches with [source? = True] [
    set intensity source-intensity ; sources
  ]
  ask patches with [wall? = True] [
    set intensity 0
    set pheromone-attract 0
  ]
end
```

```
to diffuse-chemical
  let percentage 0.95
  ; Calculate changes in intensity
  ask patches [
    let num count neighbors with [wall? = false]
    let part percentage * intensity / 8

    set intensity-change intensity-change - (num * part)
    ask neighbors with [wall? = false] [
      set intensity-change intensity-change + part
    ]
  ]

  ; Apply those changes
  ask patches [
    set intensity intensity + intensity-change
    set intensity-change 0
  ]
end

; Simulate the physics of pheromones
to simulate-pheromone
  diffuse-pheromone
  ask patches [
    set pheromone-attract (pheromone-attract * (100 - 0.1) / 100) ; evaporation
  ]
  ask patches with [wall? = True] [
    set pheromone-attract 0
  ]
end

to diffuse-pheromone
  let percentage 0.95
  ; Calculate changes in intensity
  ask patches [
    let num count neighbors with [wall? = false]
    let part percentage * pheromone-attract / 8

    set pheromone-change pheromone-change - (num * part)
    ask neighbors with [wall? = false] [
      set pheromone-change pheromone-change + part
    ]
  ]
  ; Apply those changes
  ask patches [
    set pheromone-attract pheromone-attract + pheromone-change
    set pheromone-change 0
  ]
end

;~~~~~~~~~~ patch color procedure ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
to color-patches
  ask patches [  ;; we use gray gives us value from 0 to 9.9
    let pcolor-1 to-RGB * SCALE-COLOR GRAY intensity 0 max-chemical
    let pcolor-2 to-RGB * SCALE-COLOR GRAY pheromone-attract 0 max-pheromone-attract
    let pcolor-3 to-RGB * SCALE-COLOR GRAY pheromone-repel 0 max-pheromone-repel
    set pcolor RGB pcolor-1 pcolor-2 pcolor-3
    if wall? = true [
      set pcolor white
    ]
  ]

end

;;;;;;;;;;;;;;;;;;; Reporters ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to-report distance-to-closest-source
  let closest-source min-one-of (patches with [source? = true]) [distance myself]
  report distance closest-source
end



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;; Sources ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;1) "the paper" = https://www.pnas.org/doi/10.1073/pnas.1816315116
; 2) Weber–Fechner law: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4207464/
; https://www.nature.com/articles/ncomms1455.pdf?origin=ppub -> cites textbook
; -> where you can read textbook for free:
;https://archive.org/details/PrinciplesOfNeuralScienceFifthKANDEL/page/n501/mode/2up?q=Weber–Fechner+law
; -> starts on page 451 in text. -> pg 501 in free online source
; 3) the "run" and "tumble" modes of the bacteria: https://www.cell.com/current-biology/pdf/S0960-9822(02)014
; Quantitation of the Sensory Response in Bacterial Chemotaxis: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC
```