

```

;;;;;;;;;;;;;
;;
;; le code de l'equipe rouge basique
;; préfixe : red-team
;;
;;;;;;;;;;;;;

to red-team-ask-for-energy [ b n ]
  ask b [ if (energy > 6000) [ give-energy myself n ]]
end

to red-team-go-back-to-base
  if (carrying-food? > 1000) or (energy < 500) or ((Breed = RocketLaunchers) and (nb-missiles = 0))
  [
    let b min-one-of my-bases [ distance myself ]
    if (b != nobody) [
      ifelse (distance b <= 2)
      [
        if (carrying-food? > 1000) [ give-food b carrying-food? ]
        if (energy < 500) [ red-team-ask-for-energy b 500 ]
        if ((Breed = RocketLaunchers) and (nb-missiles = 0)) [
          red-team-ask-for-energy b 500
          new-missile 5
        ]
      ]
      [
        set heading towards b - 20 + random 40
        if count turtles in-cone 2 90 with [ (breed != Burgers) and (breed != Perceptions) ] != 1 [ rt random
90 ]
        forward-move speed
      ]
    ]
  ]
end

to red-team-go-and-eat
  if ((breed = RocketLaunchers and (mem5 = 0)) or (breed = Explorers)) [
    random-move
  ]
end

to red-team-harvesters-go-and-eat
  let b min-one-of perceive-food [ distance self ]
  ifelse (b != nobody) [
    ifelse (distance b) <= 2
    [ take-food b ]
    [ set heading towards b - 20 + random 40
      if count turtles in-cone 2 90 with [ (breed != Burgers) and (breed != Perceptions) ] != 1 [ rt random
90 ]
      forward-move speed
    ]
  ]
  [ random-move ]
end

to-report red-team-no-target?
  report mem5 = 0
end

to red-team-set-target [ x y espece ]
  set mem0 x set mem1 y set mem2 espece set mem5 1
end

to red-team-call-rocket-launcher [ x y espece ]
  let rl min-one-of perceive-specific-robots color RocketLaunchers [ distancexy x y ]
  if (rl != nobody) [ ask rl [ red-team-set-target x y espece ] ]
end

to red-team-call-explorer [ x y espece ]
  let ex one-of perceive-specific-robots color Explorers
  if (ex != nobody) [
    ask ex [

```

```

        set mem0 x
        set mem1 y
        set mem2 espece
        set mem5 1
    ]
]
end

to red-team-select-target
    if (red-team-no-target?)
    [
        let h min-one-of perceive-robots ennemy [ distance myself ]
        ifelse ( h != nobody ) [
            set mem0 [xcor] of h
            set mem1 [ycor] of h
            set mem2 [breed] of h
            set mem5 1
        ]
        [ set mem5 0 ]
    ]
end

to red-team-shoot
    if (not red-team-no-target?)
    [
        ifelse (distancexy mem0 mem1 < detection-range)
        [ launch-rocket towardsxy mem0 mem1 set mem5 0]
        [ set heading towardsxy mem0 mem1 - 20 + random 40
          if count turtles in-cone 2 90 with [ (breed != Burgers) and (breed != Perceptions) ] != 1 [ rt random 90
        ]
        forward-move speed
    ]
]

end

to goRedExplorer
    red-team-go-back-to-base
    red-team-go-and-eat

    ifelse (not red-team-no-target?) [
        red-team-call-rocket-launcher mem0 mem1 mem2
        red-team-call-explorer mem0 mem1 mem2
    ]
    [ let h one-of perceive-robots ennemy
      if ( h != nobody ) [ red-team-call-rocket-launcher [xcor] of h [ycor] of h [breed] of h ]
    ]
end

to goRedRocketLauncher
    red-team-go-back-to-base
    red-team-select-target
    red-team-shoot
    red-team-go-and-eat

    if ((energy > 3000) and (nb-missiles < 5)) [ new-missile 1 ]
end

to goRedHarvester
    ;; rentre à la base si certaines conditions sont vérifiées
    red-team-go-back-to-base
    ;; cherche à manger
    red-team-harvesters-go-and-eat
end

to goRedBase
    set label energy
    if (energy > 12000) [ new-RocketLauncher self ]
    let t one-of perceive-specific-robots ennemy rocketlaunchers
    if (t != nobody) [
        launch-rocket towards t
    ]
]

```

```
end

;; procedure pour initialiser les explorers rouges
to initRedExplorer
end

;; procedure pour initialiser les rocket-launchers rouges
to initRedRocketLauncher
end

;; procedure pour initialiser les Harvesters rouges
to initRedHarvester
end

;; procedure pour initialiser les bases rouges
to initRedBase
end
```