```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;; le code de l'equipe verte basique
;; préfixe : green-team
;;
;; mem0 et mem1 : coordonnees d'une cible
;; mem5 : indique si le robot a ou non une cible sélectionnée
;; mem4 : indique si un harvester est en mode retour à la base
;; mem6 : le nb de harvesters à créer
;; mem7 : le nb de rocket-launchers à créer
;; mem8 : le nb d'explorers à créer
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

to green-team-ask-for-energy [ b n ]
  ask b [ if (energy > 1000) [ give-energy myself n ]]
end

;;
;; rentre à la base si le robot transporte plus de 1000 unités de nourriture ou si son énergie est inférieure à
500 ou si c'est un rocket-launcher qui n'a plus de missiles
;;
to green-team-go-back-to-base
  ;; détermine la base la plus proche
  let b min-one-of my-bases [ distance myself ]
  if (b != nobody) [
    if ((breed = Harvesters) and (distance b > 10) and (distance b < 11)) [drop-wall]
    ;; si le robot est arrivé à la base
    ifelse (distance b <= 2)
      [
        ;; dépose sa nourriture
        give-food b carrying-food?
        ;; demande de l'énergie si il reste moins de 1000
        if (energy < 1000) [ green-team-ask-for-energy b 300 ]
        ;; demande de l'énergie pour créer 5 missiles
        if ((Breed = RocketLaunchers) and (nb-missiles = 0)) [
          green-team-ask-for-energy b 500
          new-missile 5
        ]
        set mem4 0
        ;; fait demi-tour
        rt 180
      ]
    ;; sinon
      [
        ;; s'oriente vers la base avec un peu d'aléatoire
        set heading towards b - 20 + random 40
        ;; fait un demi tour si il y a du monde devant
        ifelse (free-ahead? speed = nobody)  [ forward-move speed ][ rt random 360 if (free-ahead? 2 = nobody)
[ forward-move speed ]]
      ]
    ]
end

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

to green-team-harvesters-go-and-eat
  let w min-one-of (perceive-walls) [distance myself]
  let b min-one-of my-bases [ distance myself ]
  if ((w != nobody) and ((distance b < 10) or (distance b > 11))) [take-wall w]
  let f min-one-of perceive-food [ distance myself ]
  ifelse (f != nobody) [
    ifelse (distance f) <= 2
      [ take-food f ]
      [ set heading towards f - 20 + random 40
      ifelse (free-ahead? speed = nobody) [ forward-move speed ][ rt random 360 if (free-ahead? 2 = nobody) [
forward-move speed ]]
      ]
  ]
```

```
  [
    ifelse (mem5 = 1)
    [
      ifelse (distancexy mem0 mem1 > 1)
      [
        set heading towardsxy mem0 mem1
        ifelse (free-ahead? speed = nobody) [ forward-move speed ][ rt random 360 if (free-ahead? 2 = nobody) [
forward-move speed ]]
      ]
      [ set mem5 0 ]
    ]
    [ random-move ]
  ]
end

;;
;; la mémoire mem5 est à
;; - 0 si le robot n'a pas de cible sélectionnée
;; - 1 si le robot a une cible sélectionnée
;;
to-report green-team-no-target?
  report mem5 = 0
end

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

to green-team-set-target-t [ t espece ]
  set mem0 t set mem2 espece set mem5 2
end

to green-team-set-food-target [ x y ]
  if (green-team-no-target?) [ set mem0 x set mem1 y set mem5 1 ]
end

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

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

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

to green-team-call-harvester [ x y ]
  let h min-one-of perceive-specific-robots color harvesters [ distancexy x y ]
  if (h != nobody) [ ask h [ green-team-set-food-target x y ]]
end

;;
;; essaye de sélectionner une cible à viser
;;
to green-team-select-target
  ;; si le robot n'a pas une base adverse comme cible
  if (green-team-no-target?)
  [
    ;; il essaye de percevoir un robot ennemi (le plus proche de lui)
    let h min-one-of perceive-robots ennemy [ distance myself ]
    ;; si il en a vu un, il mémorise ses coordonnées et vérouille la cible
    ifelse ( h != nobody ) [
      set mem0 h
      set mem2 [breed] of h
      set mem5 2
```

```
    ]
    ;; sinon, il se dévérouille
    [ set mem5 0 ]
  ]
end

to green-team-shoot
  if (not green-team-no-target?)
  [
    ifelse (mem5 = 1)
    [ launch-rocket towardsxy mem0 mem1 set mem5 0 ]
    [ launch-faf mem0 set mem5 0 ]
  ]

end

to green-team-drive-harvesters
  let food one-of perceive-food
  if (food != nobody) [
    green-team-call-harvester [xcor] of food [ycor] of food
  ]
end

to goGreenExplorer
;  ifelse (carrying-food? > 1000) or (energy < 500) or ((Breed = RocketLaunchers) and (nb-missiles = 0))
  ifelse (energy < 1000)
  [ green-team-go-back-to-base ]
  [ green-team-go-and-eat ]

  green-team-drive-harvesters

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

to goGreenRocketLauncher
;  ifelse (carrying-food? > 1000) or (energy < 500) or ((Breed = RocketLaunchers) and (nb-missiles = 0))
  ifelse (energy < 1000) or (nb-missiles = 0)
  [
    ;; rentre à la base si certaines conditions sont vérifiées
    green-team-go-back-to-base
  ]
  [
    ;; sinon sélectionne une cible
    green-team-select-target
    ifelse (green-team-no-target?)
    ;; si pas de cible, cherche à manger
    [ green-team-go-and-eat ]
    ;; sinon tire
    [ green-team-shoot ]
  ]

  ;; crée un nouveau missile si plus de 3000 unités d'énergie
  if ((energy > 3000) and (nb-missiles < 5)) [ new-missile 1 ]
end

to goGreenHarvester
  let f min-one-of perceive-food [ distance myself ]
  while [(f != nobody) and (distance f <= 2)] [
    take-food f
    set f min-one-of perceive-food [ distance myself ]
  ]

  ifelse (mem4 = 1) or (carrying-food? > 500) or (energy < 100)
  [
    ; retour à la base
    set mem4 1
```

```
    ;; rentre à la base si certaines conditions sont vérifiées
    green-team-go-back-to-base
    if (energy > 100) and (carrying-food? > 100) [
      ;; détermine la distance de la base la plus proche
      let b min-one-of my-bases [ distance myself ]
      if (b != nobody) [
        ;; si le robot est à moins de 10 de la base
        if (distance b < 10)
          [ plant-seeds color max-seeds ]
      ]
    ]
  ]
  [
    ;; cherche à manger
    green-team-harvesters-go-and-eat
  ]
end

to goGreenBase
  ;; crée un nouveau robot si il reste des demandes de création dans le pipe (mem6, mem7 ou mem8)
  ifelse (mem6 > 0) [ new-Harvester self set mem6 mem6 - 1 ]
  [ ifelse (mem7 > 0) [ new-RocketLauncher self set mem7 mem7 - 1 ]
  [ if (mem8 > 0) [ new-Explorer self set mem8 mem8 - 1 ]]]

  if (nb-missiles < 10) and (energy > 1000) [ new-missile 50 ]
  if (nb-fafs < 10) and (energy > 1000) [ new-faf 10 ]
  if (energy > 12000) [ ifelse (random 2 = 0) [ set mem6 mem6 + 1 ][ifelse (random 2 = 0) [ set mem7 mem7 + 1 ]
[ set mem8 mem8 + 1 ]]]

  let t one-of perceive-specific-robots ennemy rocketlaunchers
  if (t != nobody) [
    ;; regarde s'il y a des robots amis en direction de la cible potentielle t avant de tirer
    if (not any? perceive-robots-in-cone color towards t) [ launch-faf t ]
  ]
end

;; procedure pour initialiser les explorers verts
to initGreenExplorer
end

;; procedure pour initialiser les rocket-launchers verts
to initGreenRocketLauncher
end

;; procedure pour initialiser les harvesters verts
to initGreenHarvester
end

;; procedure pour initialiser les bases verts
to initGreenBase
  ; crée des explorers, des harvesters et des rocket-launchers
  new-Harvester self
  set mem6 7    ;; comme on ne peut créer qu'un seul robot par tour, mem6 sert à savoir combien de harvesters
on doit encore créer
end
```