

CHANGES of BEE-STEWARD code in comparison to Bumble-BEEHAVE in "Go-Procedures"

This document alphabetically lists all procedures involved in the "Go"-process and the changes made in comparison to Bumble-BEEHAVE. An overview which procedures have been changed can be found in the document "Scheduling_BEE-STEWARD.xlsx" (sheet: "Comparison to BBH Go Procedures").

to ActivityProc

```
let break_s 0.5 * 3600 ; time a bee spends resting
set activity "resting"
if colonyID >= 0 ; only colony members can engage in tasks
[
  ask Colony colonyID ; the stimuli in a bees' colony are determined
  [
    set stimEgglaying StimEgglayingREP
    set stimNectarForaging StimForagingNectarREP
    set stimPollenForaging StimForagingPollenREP
    set stimNursing StimNursingREP
  ]

  ; if a colony-specific stimulus exceeds the individual threshold, "activity" of the bee is set to this
  ; particular task,
  ; tasks are ordered by their importance:
  if [ stimEgglaying ] of Colony colonyID > thEgglaying [ set activity "egglaying" ]
  if [ stimNursing ] of Colony colonyID > thNursing [ set activity "nursing" ]
  if [ stimPollenForaging ] of Colony colonyID > thForagingPollen [ set activity "pollenForaging" ]
  if [ stimNectarForaging ] of Colony colonyID > thForagingNectar [ set activity "nectarForaging" ]
]

if speciesName = "Psithyrus" and (activity = "nursing" or activity = "pollenForaging" or activity =
"nectarForaging")
[ set activity "resting" ]

if activity = "resting" [ set activityList lput "REST" activityList
  set personalTime_s personalTime_s + break_s
]
if activity = "egglaying" [ set activityList lput "EGG" activityList
  EgglayingProc
]
if activity = "nursing" [ set activityList lput "NURSE" activityList
  BroodIncubationProc
]
if activity = "pollenForaging"
[ set activityList lput "P-FOR" activityList
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
        ForagingProc
      ]
    if activity = "nectarForaging"
      [ set activityList lput "N-FOR" activityList
        ForagingProc
      ]
    end

; *****
```

to AssertionProc [message]

```
  show message
  set AssertionMessage message
  set AssertionViolated true
end

; *****
```

to BadgersOnTheProwlProc

```
  let foragingRange_m 735 ; estimated from Kruuk & Parish, J. Zool., Lond. (1982) 196,31-39, Tab. 1:
  territory: ca. 170ha, hence radius ca. 735m
  let encounterProb 0.19 ; probablity to come across the nest; Kowalczyk et al 2006, Wildlife Biology
  12(4):385-391. 2006 Tab1; DR% daily range as % of total home range: 19% (19+-18%);
  let digUpProb 0.1 ; probability to perceive the nest and dig it up - ARBITRARY VALUE
  ask Badgers
  [
    let memoX xcor
    let memoY ycor
    ask colonies with [ distancexy memoX memoY < SCALING_NLpatches/m * foragingRange_m ]
    [
      if random-float 1 < encounterProb * digUpProb
      [
        set energyStore_kJ 0
        set pollenStore_g 0
        set color red
        let victimColonyID who
        let memobroodDeaths 0
        ask bees with [ colonyID = victimColonyID ]
        [
          if brood? [set memobroodDeaths memobroodDeaths + number]
          DieProc "Colony killed by badger!"
        ]
        set broodDeathBadger broodDeathBadger + memobroodDeaths
      ]
    ]
  ]
end

; *****
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

to BroodIncubationProc

; Heinrich p. 70, Fig. 5.2: 144 temperature recordings in 24hrs (every 10 minutes) of which ca. 48 are below optimal (20 heating periods)

; hence: $96 * 10 \text{ min} = 960 \text{ mins}$ heating in 20 periods = 2880s heating

; Silvola 1984: Queen (*B. terrestris*) spends ca. 10kJ/day for incubation

; *Bombus terrestris* Queen weight: mean: 0.8g (Beekman et al 1998) (2017-03-29: HOWEVER:

Silvola assumes a weight of 650mg, which should have been used here)

; hence: $10\text{kJ}/0.8\text{g} = 12.5 \text{ kJ/g}$ per day provided from heating bees

; with 1 day = 960 minutes (16hrs) (Heinrich (see above))

; hence incubation energy per heating bee-mass is:

; 0.0130208333kJ per minute and g (or 0.000217013888 kJ per second and g)

; or 0.000000217013888 kJ per second and mg

let heatingPeriod_s 2880 ; (2880s = 48 min) time spent on incubation - ca. 48 min. between foraging flights of incubating queen, Heinrich, p. 92, Fig. 5.2

let incubationEnergy_kJ_per_mg_s 0.000000217013888 ; (better alternative: 0.00000026709?) kJ per mg heating bee-mass per second (calculation see above)

let heatProvided_kJ heatingPeriod_s * incubationEnergy_kJ_per_mg_s * weight_mg * number ; [kJ] energy released by heating bee during an incubation phase

let heatProvidedPerBrood_kJ 0

if [allEggs + allLarvae + allPupae] of Colony colonyID > 0

[set heatProvidedPerBrood_kJ heatProvided_kJ / [allEggs + allLarvae + allPupae] of Colony colonyID] ; the amount of heat is equally distributed over the brood

let memoColonyID colonyID

ask bees with [colonyID = memoColonyID and (stage = "egg" or stage = "larva" or stage = "pupa")]

[set cumullIncubationReceived_kJ cumullIncubationReceived_kJ + heatProvidedPerBrood_kJ]

ask Colony colonyID

[

set summedIncubationToday_kJ summedIncubationToday_kJ + heatProvided_kJ

set energyStore_kJ energyStore_kJ - heatProvided_kJ ; energy spent for heating is subtracted from colonies' energy stores

]

set personalTime_s personalTime_s + heatingPeriod_s ; heating takes some time..

end

; *****

to CheckNumbersProc

let totalInitialBees 0

B_lapidarius

+B_pascuorum

+foreach BeeSpeciesInitialQueensList ; e.g. ["B_hortorum" 100] ["B_terrestris" 500]]

+B_hortorum

+B_hypnorum

+B_pratorum

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

→ N_Psithyrus

[set totalInitialBees totalInitialBees + round (item 1 ? * InitialQueensModifierBehaviorSpace)]

let totalBeesPresent sum [number] of bees

if totalInitialBees + TotalBeesEverProduced - TotalBeesEverDied != totalBeesPresent
[AssertionProc "Error in number of bees ever produced/died! (CheckNumbersProc)"]

set TotalEggs sum [allEggs] of Colonies
set TotalLarvae sum [allLarvae] of Colonies
set TotalPupae sum [allPupae] of Colonies
set TotalAdultWorkers sum [allAdultWorkers] of Colonies
set TotalAdultQueens sum [number] of bees with [stage = "adult" and caste = "queen"] ; total
adult queens, (not necessarily member of a colony!)
set TotalAdultMales sum [allAdultMales] of Colonies
set TotalAdults sum [number] of bees with [stage = "adult"] ; total adults, including
queens (not necessarily member of a colony!)
set TotalForagingTripsToday 0

if TotalEggs != sum [number] of bees with [stage = "egg"]
or TotalLarvae != sum [number] of bees with [stage = "larva"]
or TotalPupae != sum [number] of bees with [stage = "pupa"]
or TotalAdultWorkers != sum [number] of bees with [caste = "worker" and stage = "adult"]
or TotalAdultMales != sum [number] of bees with [stage = "adult" and caste = "male"]
[AssertionProc "Error in total numbers of bees! (CheckNumbersProc)"]

if day = 364

[
let colCheck TotalColoniesEverProduced - (ColonyDeathsEndSeason + ColonyDeathsNoBees)
if colCheck != 0 [AssertionProc "Colonies produced cannot be accounted for"]
ask deadCols

[
let eggcheck totalEggsProduced - (totalAdultsProduced
+ broodDeathsEnergyStores
+ broodDeathsNoAdults
+ broodDeathsCP
+ broodDeathEndSeason
+ broodDeathBadger
+ eggDeathsIncubation
+ larvaDeathsIncubation
+ larvaDeathsWeight
+ pupaDeathsIncubation
)

if eggcheck != 0 [AssertionProc (word "Dead Colony " who " cannot account for all eggs produced"
eggcheck)]

]

]

end

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
, *****
```

to-report CompetitionPointDateREP

```
; determines the date of a colonies' competition point
let compDate NotSetHigh
let x queenProductionDate - eusocialPhaseDate ; range: ca; -5..40 i.e. y ca. 12 - 43.5
let y 0.7 * x + 15.5 ; from Duchateau & Velthuis 1988, Fig. 6
let latestCPafter_d 45
set compDate round (eusocialPhaseDate + y)
if compDate - eusocialPhaseDate > latestCPafter_d
  [ set compDate eusocialPhaseDate + latestCPafter_d ]
report compDate
end
```

```
, *****
```

to CreateColoniesProc

; Purpose: creates and sets up initial values of a newly founded colony

```
let whoColony -1
let memoSpeciesID -1
let memoSpeciesName ""
let xcol -1
let ycol -1
let speciesShape "bumblebeenest" ; will be replaced by species specific bumblebee symbol
let nNewColonies count bees with [ activity = "nestConstruction" ]

; each colony needs 2 storebars to display on the interface the relative amount of nectar and pollen
stored:
create-storebars 2 * nNewColonies
[
  set shape "halfline"
  set heading 90
  set size 10 * MasterSizeFactor
  set maxSize size
  set storeColonyID whoColony
]

; now the COLONIES are created:
create-Colonies nNewColonies
[
  set whoColony who ; the ID of the colony
  set colonyFoundationDay ticks
  ask one-of bees with [ activity = "nestConstruction" ]
  [
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

set xcol xcor ; x and y coordinates of the queen are saved, so that the nest can be located where the queen is

set ycol ycor

set colonyID whoColony ; queen gets the ID of the colony..

set memoSpeciesID SpeciesID ; and saves her species-type for the colony

[set memoSpeciesName speciesName](#)

set activity "resting" ; as the colony is created now, the queen rests

set speciesShape speciesName ; saves the species of the queen so that the colony can be displayed in the according shape

if ShowQueens? = true [show-turtle]

]

set queenProduction? false ; no production of queens yet

set switchPointDate NotSetHigh ; queen won't lay haploid eggs until switchPointDate is re-set

set competitionPointDate NotSetHigh

set eusocialPhaseDate NotSetHigh

set queenProductionDate NotSetHigh

set speciesIDcolony memoSpeciesID ; colony gets species-type from queen

[set speciesNameColony memoSpeciesName](#)

if ShowNests? = false [hide-turtle]

set xcor xcol ; the colony is placed at the location of queen

set ycor ycol

set queenright? true ; queen is still alive

set shape speciesShape ; colony is displayd on the interface as a bumblebee, showing the species of the queen

set heading 0

set color 33 ; (33) dark brown

set size ColonySymbolsize; 14 (14)

set energyStore_kJ 100 * EnergySucrose_kJ/mymol * 1.5 ; i.e. 0.873kJ (= 100 microliter of 1.5M nectar (i.e. ca. 1 crop))

set colonysize 1 ; i.e. the queen

set cohortBased? true

if count Colonies with [cohortBased? = false] < COLONIES_IBM

[

set cohortBased? false

set color ColorIBM

set InspectTurtle Who

]

PatchesInRangeProc

FoodsourcesInFlowerAndRangeProc

let barX 3.5

let barY 5

; a nectar and a pollen storebar is now assigned to the new colony

ask one-of storebars with [storeColonyID = -1]

[

ifelse xcol - barX > min-pxcor and ycol - barY > min-pycor

[setxy xcol - barX ycol - barY]

[hide-turtle]

set storeColonyID whoColony

set store "Nectar"

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    set color yellow
  ]

  set barY barY - 1
  ask one-of storebars with [ storeColonyID = -1 ]
  [
    ifelse xcol - barX > min-pxcor and ycol - barY > min-pycor
    [ setxy xcol - barX ycol - barY ]
    [ hide-turtle ]
    set storeColonyID whoColony
    set store "Pollen"
    set color orange - 0.5
  ]
  set TotalColoniesEverProduced TotalColoniesEverProduced + 1
]
end

```

```

, *****
;

```

to CreateFoodsourcesProc

```

ifelse ( file-exists? InputMapMyMap )
[
  import-pcolors InputMapMyMap
  ask patches [ set pcolorSave pcolor ]
]
[
  ask patches
  [
    set pcolor 5Backgroundcolor ; Backgroundcolor = 5 background/matrix colour if no map image is
    available; color 5 = grey
    set pcolorSave pcolor
  ]
]

ifelse ( file-exists? FoodsourcesFileINPUT\_FILE )
[
  file-open FoodsourcesFileINPUT\_FILE
  set SCALING_NLpatches/m precision (1 / file-read) 8 ; CAUTION! Scaling in BEESCOUT:
  m/NLpatch; Scaling Bumbleworld: NLpatches per m !!!
  let dustbin file-read-line ; N patches in old input file format or heading in new format
  if length dustbin <= 10 [ set dustbin file-read-line ] ; heading
  while [ not file-at-end? ]
  [
    create-foodsources 1
    [
      ; imported file format:

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

; ;;; OLD: id patchType patchColour xcor ycor size_sqm quantityPollen_g quantityNectar_l
;concentration startDay stopDay corollaLength_mm nectarFlowerVolume_myI interFlowerTime_s
;patchInfo
_____ ; id patchType patchColour xcor ycor size_sqm flowerSpeciesList perimeter_m info
set id_Beescout file-read ; column 1: id
set patchType file-read ; column 2: patchType
set flowerSpecies_relativeAbundanceList (list patchType 1)
let memoFoodpatchColour file-read ; column 3: patchColour: the colour of the food patch, as
shown on the map
set color memoFoodpatchColour - 1 ; the colour of the food source (= turtle), slightly darker
then the food patch to be visible
set colorMemo color ; saves original color (for use in buttons)
set xcor file-read ; column 4: xcor
set ycor file-read ; column 5: ycor
set area_sqm file-read ; column 6: area_sqm [m^2]
_____ set pollen_g AbundanceBoost * file-read ; [g]
_____ set pollenMax_g pollen_g
_____ set proteinPollenProp file-read
_____ set nectar_myI AbundanceBoost * file-read * 1000 * 1000 ; [quantityNectar_l: l * 1000 = ml; ml
* 1000 = myI]
_____ set nectarMax_myI nectar_myI
_____ set nectarConcentration_mol/l file-read ; [mol/l]
_____ set startDay file-read ; day of year
_____ set stopDay file-read ; day of year
_____ set corollaDepth_mm file-read ; [mm]
_____ set nectarFlowerVolume_myI file-read ; [microlitre]
_____ set interFlowerTime_s file-read ; [s]
set flowerSpeciesList file-read ; column 7: flowerSpeciesList ; [s]
_____ set perimeter_m file-read ; column 8: perimeter_m NEW FOR BEESTEWARD!
_____ set occurrence "patch" ; NEW FOR BEESTEWARD!
set patchInfo file-read-line ; column 9: patchInfo: the rest of the line is now read in
set radius_m sqrt (area_sqm / pi) ; [m]
set shape "circle"
set size FoodsourceSymbolSizeFactor * radius_m * Scaling_NLpatches/m
if size < MinFoodSourceSymbolSize [ set size MinFoodSourceSymbolSize ]
ifelse ShowFoodsources? = false
[ hide-turtle ]
[ show-turtle ]
set masterpatch? true
set layersInPatchList (list who)
set masterpatchID who
_____ set farmland? false
_____ if member? (word " " who " ") MyFarmlandPatches [ set farmland? true ]
_____ set stewardshipSpeciesList []
if MapAreaIncluded = "top left quarter" and (xcor > max-pxcor / 2 or ycor < max-pycor / 2) [
die ]
if MapAreaIncluded = "top right quarter" and (xcor <= max-pxcor / 2 or ycor < max-pycor / 2) [
die ]
if MapAreaIncluded = "bottom left quarter" and (xcor > max-pxcor / 2 or ycor >= max-pycor / 2)
[ die ]

```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    if MapAreaIncluded = "bottom right quarter" and (xcor <= max-pxcor / 2 or ycor >= max-pycor / 2) [ die ]
    if MapAreaIncluded = "top half" and (ycor < max-pycor / 2) [ die ]
    if MapAreaIncluded = "bottom half" and (ycor >= max-pycor / 2) [ die ]
    if MapAreaIncluded = "left half" and (xcor > max-pxcor / 2) [ die ]
    if MapAreaIncluded = "right half" and (xcor <= max-pxcor / 2) [ die ]
  ]
]
file-close
]
[
  user-message (word "TheThere is no such FoodsourcesFileINPUT_FILE (""FoodsourcesFileINPUT_FILE ") does not exist" in the current directory!;!)"
  stop
]

```

```

if MergeHedges? = true [_MergeHedgesProc_]

```

```

  ifelse CircularBrush? = false
  [ ; square brush
    set BrushArea_ha (BrushSize / Scaling_NLpatches/m) ^ 2 / 10000
  ]
  [ ; circular brush
    set BrushArea_ha (pi * ((BrushSize / Scaling_NLpatches/m) / 2) ^ 2) / 10000
  ]

```

```

CreateLayersProc ; creates new foodsources from those foodsources with multiple species (i.e.
with flowerSpeciesList != [] )
  set TotalFoodSources count foodsources
end

```

```

, *****
,

```

to CreateLayersProc

; if the flowerSpeciesList of food source is not empty (i.e. it usually contains several plant species that might be in flower at different times) this procedure then creates a single flowerspecies food source (at the same location, area etc) for
; each flowerspecies of the original foodsource. At the end, the original food source is removed.

; Remove foodsources with low resource values

; If the switch MinSizeFoodSources? is ON, all foodsources with either nectarMax or pollenMax values under a certain threshold will

; have that resource set to 0. This is to prevent foragers from visiting low-resource flowers and having very high handling times, leading

; to poor colony performance. The thresholds for nectar and pollen are set below, each one being the amount of nectar/pollen an average Bterr

; queen can carry multiplied by the FoodSourceLimit interface variable. For example, if the variable is set to 20, the minimum nectar/pollen

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWART: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

; amount at a foodsource is enough for 20 trips by a *B. terrestris* queen with a crop size of 180myl and pollen pellets of 0.05g.

; If the switch RemoveEmptyFoodSources? is ON, all foodsources with BOTH nectarMax and pollenMax levels set to 0 are removed from

; the model, which greatly improves the speed of the model.

; set minimum resource limits for foodsources. FSs with lower nectar or pollen amounts than these will have their values set to 0 if SetEmptySmallFoodSources? is TRUE

let foodsourcesRemoved false

let minNectSize_myl FoodSourceLimit * 180 ; minimum nectar for a foodsource is amount equal to "FoodSourceLimit" number of trips by a queen with a crop volume of 180myl

let minPolsize_g FoodSourceLimit * 0.05 ; minimum pollen for a foodsource is amount equal to "FoodSourceLimit" number of trips by a queen with pollen pellet size of 0.05g

ifelse (file-exists? FlowerspeciesFile)

[
let flowerspeciesDataCSV csv:from-file FlowerspeciesFile ; reads flower species data from csv file and saves it in list, i.e. [[line 1][line 2]..[last line]]

let header item 0 flowerspeciesDataCSV ; saves header = first line of csv file = item 0 of list
; e.g. ["Flowerspecies" "Default_pollen_g/m2" "Default_nectar_ml/m2" "proteinPollenProp" "concentration_mol/l" "startDay" "stopDay" "corollaDepth_mm" "nectarFlowerVolume_myl" "intFlowerTime_s"]

let allFlowerspeciesList []

foreach but-first flowerspeciesDataCSV ; but-first: ignores header

[
let flowerSpec read-from-string item 0 ? ; gets the first value (= flower species) of each column (in actual order)

set allFlowerspeciesList lput flowerSpec allFlowerspeciesList ; the species is now added to the list containing all possible flower species

; e.g. ["Bugle" "Burdock" "Oilseed_rape" "Giant_bindweed" "Common_knapweed" ...]

]

ask foodsources with [flowerSpeciesList != []] ; if foodsource is composed of several flower species

[
let memoMasterpatchID -1

let currentoccurrence "patch"

foreach flowerSpeciesList ; for each flowerspecues a new foodsource is created
(flowerSpeciesList is a Foodsources-own)

[
ifelse member? item 0 ? CSS_OptionsList

[set currentoccurrence item 0 ?]

]

— hatch 1

— [

— set flowerSpecies_relativeAbundanceList ? ; e.g. ["Bugle" 0.236]

— let mySpecies item 0 flowerSpecies_relativeAbundanceList ; e.g. "Bugle"

— if member? "Margin" mySpecies

— let myDataLine item (position mySpecies allFlowerspeciesList + 1) flowerspeciesDataCSV

— ; myDataLine: the relevant line of the csv file for this particular flower species; position..+1 to account for header

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

__ ; e.g. ["\"Bugle\"" 7.7381E-5 9.63889E-5 0.072103857 0.824737635 120 211 10 0.80966667
2.5]
__ let myRelativeAbundance AbundanceBoost * (item 1 flowerSpecies_relativeAbundanceList) ;
proportion of patch area covered by this species
__ ; pollen available at patch: pollen produced by this plant species per m2 (where present) *
total area of this foodsource * relative abundance of this flowerspecies in the habitat *
ProportionModelledColonies (as only a fraction of the colonies is simulated)
__ set pollenMax_g area_sqm * myRelativeAbundance * (item (position "pollen_g/flower"
header) myDataLine)
__ set nectarFlowerVolume myl (item (position "nectar_ml/flower" header) myDataLine) * 1000
__ set nectarMax_myl area_sqm * myRelativeAbundance * nectarFlowerVolume myl
__ set nectarConcentration_mol/l (item (position "concentration_mol/l" header) myDataLine)
__ set proteinPollenProp (item (position "proteinPollenProp" header) myDataLine)
__ set startDay (item (position "startDay" header) myDataLine)
__ set stopDay (item (position "stopDay" header) myDataLine)
__ set corollaDepth_mm (item (position "corollaDepth_mm" header) myDataLine)
__ set nectarFlowerVolume_myl (item (position "nectarFlowerVolume_myl" header) myDataLine)
__ set interFlowerTime_s (item (position "intFlowerTime_s" header) myDataLine)

__ if member? (word " " who " ") MyFarmlandPatches [ set farmland? true ]
__ ; Set nectar / pollen levels to 0 if smaller than the minimum size
__ if MinSizeFoodSources? AND nectarMax_myl < minNectSize_myl
__ [ set nectarMax_myl 0 ]
__ if MinSizeFoodSources? AND pollenMax_g < minPolSize_g
__ [ set pollenMax_g 0 ]

__ ; kill the foodSource if both nectar and pollen are below the respective minimum values and
if RemoveEmptyFoodSources? is TRUE
__ if nectarMax_myl = 0 AND pollenMax_g = 0 AND RemoveEmptyFoodSources?
__ {
__ AND member? "Crop " patchType = false ; NEW FOR BEESTEW: crops are
not removed to allow having e.g. margins around cereal fields
__ [
__ set foodsourcesRemoved true
__ die
__ ]

__ ifelse memoMasterpatchID < 0 ; if the masterpatch hasn't been set yet..
__ [
__ set memoMasterpatchID who ; ..the first foodsource/layer will be the masterpatch
__ set masterpatchID memoMasterpatchID ; only masterpatchID has to be updated, as
'masterpatch?' is true by default
__ ]
__ [
__ set masterpatch? false ; .. for all other 'layers' of the original foodsource, masterpatch? is
set false
__ set masterpatchID memoMasterpatchID ; set to the first 'layer' created at this flower patch

__ set layersInPatchList [] ; will be populated later
__ if ShowMasterpatchesOnly? = true [ hide-turtle ] ; non-masterpatches might be hidden
__ ]

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    _]
  ]
] ; end of "foreach flowerSpeciesList" loop
die ; the original foodsource is no longer needed and can be removed
]
]
[ if FlowerspeciesFile != "No Input File"
  [ user-message "There is no such FlowerspeciesFile in current directory!" ]
]
; if foodsourcesRemoved = true [ output-print "One or more very small food sources removed! To
avoid, set RemoveEmptyFoodSources? 'false'!" ]

```

```

; set layersInPatchList to a list of all foodsources at same location
ask foodsources [set layersInPatchList sort [who] of foodsources here]
[
  set layersInPatchList sort [who] of foodsources here
  if occurrence = "margin"
  [
    set shape "fieldmargin2" ; margins of (crop) fields are presented on the map as a blue "M"
    ask foodsource masterpatchID [ set shape "fieldmargin2" ]
  ]
  if occurrence = "plot"
  [
    set shape "fieldplot" ; "plot" is represented by a blue "P"
    ask foodsource masterpatchID [ set shape "fieldplot" ]
  ]
  if occurrence = "legume"
  [
    set shape "fieldlegume" ; "legumes" are represented by a blue "L"
    ask foodsource masterpatchID [ set shape "fieldlegume" ]
  ]
]
]

; Update total CSO areas:
ask foodsources with [ masterpatch? = true ] ; and not member? "no info" patchInfo ]
[
  let memoWho who
  if any? foodsources with [masterpatchID = memoWho and occurrence = "margin"]
  [
    let myCSOarea m2 read-from-string patchInfo
    if is-string? myCSOarea m2 [ set myCSOarea m2 read-from-string myCSOarea m2 ]
    set CSS TotalAreaMargin ha CSS TotalAreaMargin ha + myCSOarea m2 / 10000
    set CSS TotalAreaAll ha CSS TotalAreaAll ha + myCSOarea m2 / 10000
  ]

  if any? foodsources with [masterpatchID = memoWho and occurrence = "plot"]
  [
    let myCSOarea m2 read-from-string patchInfo
    if is-string? myCSOarea m2 [ set myCSOarea m2 read-from-string myCSOarea m2 ]
    set CSS TotalAreaPlot ha CSS TotalAreaPlot ha + myCSOarea m2 / 10000
    set CSS TotalAreaAll ha CSS TotalAreaAll ha + myCSOarea m2 / 10000
  ]
]

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
_]
if
any? foodsources with [masterpatchID = memoWho and occurrence = "legume"]
_]
let myCSOarea m2 read-from-string patchInfo
if is-string? myCSOarea m2 [ set myCSOarea m2 read-from-string myCSOarea m2 ]
set CSS TotalAreaLegume ha CSS TotalAreaLegume ha + myCSOarea m2 / 10000
set CSS TotalAreaAll ha CSS TotalAreaAll ha + myCSOarea m2 / 10000
_]
_]
end

. *****
,
```

to-report CropAndPelletSizeREP [forage]

```
; bee crop and pollen capacity based on weight using (1) linear formula
; (2) pollen:crop ratio from HBs in BEEHAVE, with both
; having an upper limit set by species-own variables.
let beeWeightToLoadFactor 0.402 ; 0.402: derived from Ings et al 2006 for nectar loads
let beehaveCropToPelletFactor_ul-to-g 0.015 / 50 ; nectar load BEEHAVE: 50ul (Winston (1987),
Nuñez (1966, 1970), Schmid-Hempel et al. (1985); POLLENLOAD 0.015 [g] (from HoPoMo, Schmickl
Crailsheim 2007, based on Seeley 1995)
let result 0
let maxCropVol_myl [ specMax_cropVolume_myl ] of oneSpecies speciesID
let maxPollen_g [ specMax_pollenPellets_g ] of oneSpecies speciesID
if forage = "nectar"
[ set result min list (maxCropVol_myl) (weight_mg * beeWeightToLoadFactor) ] ; result is the
lower of these two values
; equation derived from Ings et al 2006
if forage = "pollen"
[ set result min list (maxPollen_g) (weight_mg * beeWeightToLoadFactor *
beehaveCropToPelletFactor_ul-to-g) ] ; result is the lower of these two values
; pollen load in same proportion as values used in BEEHAVE
if result = 0 [ AssertionProc "Assertion violated: Error in CropAndPelletSizeREP" ]
report result ; units: ul for nectar, g for pollen!
end
```

```
, *****
,
```

to CropRotationProc

```
let year ceiling (ticks / 365)
let nMaps length CropRotationList
let mapToUse remainder year nMaps
set FoodsourcesFile item mapToUse CropRotationList
ask foodsources [ die ]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
ask bees[
  set knownMasterpatchesNectarList []
  set knownMasterpatchesPollenList []
  set nectarsourceToGoTo -1
  set pollensourceToGoTo -1
  set currentFoodsource -1

]
ask colonies
[
  set masterpatchesInRangeList []
  set masterpatchesWithNectarlayersInFlowerAndRangeList []
  set masterpatchesWithPollenlayersInFlowerAndRangeList []
  set nectarInFlowerAndRangeList []
  set pollenInFlowerAndRangeList []
]
CreateFoodsourcesProc
ask species
[
  ;Create list of foodSources as nest sites and calculate their total area
  set nestsiteFoodsourceList FoodSources with [ (member? patchtype [nestHabitatsList] of myself)
AND masterPatch? ]

  set nestSiteArea sum [area sqm] of nestsiteFoodsourceList
  ; queen may start egg lying once 50% of pollen needed to raise 1 batch of eggs is stored:
  set minPollenStore g 0.5 * 0.001 * devWeightPupationMin mg * batchsize /
pollenToBodymassFactor
  if count nestsiteFoodsourceList = 0 and name != "Psithyrus"
  [output-print (word name " has no suitable nesting habitat")]
]

if Report_name != 0 ; in this case, CropRotationProc was called by ButtonGenerateOutputProc
[ file-open ( word Report_name ".csv") ]
end

. *****
/
```

to-report DateREP

```
let month-names (list "January" "February" "March" "April" "May" "June" "July" "August"
"September" "October" "November" "December")
let days-in-months (list 31 28 31 30 31 30 31 31 30 31 30 31)
let year floor (ticks / 365.01) + 1
if ticks = 0 [ set year 0 ]
let month 0
let dayOfYear remainder ticks 365
if dayOfYear = 0 [ set dayOfYear 365 ]
let dayOfMonth 0
let sumDaysInMonths 0
while [ sumDaysInMonths < dayOfYear ]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
[
  set month month + 1
  set sumDaysInMonths sumDaysInMonths + item (month - 1) days-in-months
  set dayOfMonth dayOfYear - sumDaysInMonths + item (month - 1) days-in-months
]
let result ""
if month > 0
  [ set result (word dayOfMonth " " (item (month - 1) month-names) " " year ) ]
report result
end
```

, *****

to-report DetectionProbREP [patchWho xcol ycol]

```
let dist_m 0
let patchRadius_m -999
; calculation of the distance between Foodsource and Colony:
ask foodsource patchWho
[
  set dist_m (distancexy xcol ycol) / SCALING_NLpatches/m ; distancexy: in NetLogo gridcells
  set patchRadius_m radius_m
]
; calculation of the detection probability, based on the distance (see BEESCOUT model, Becher et al.
2016, Ecological Modelling):
let relevantDistance_m dist_m - patchRadius_m ; the "relevant" distance is the distance to the edge
of the field, i.e. dist_m (=centre) - patchRadius_m
if relevantDistance_m < 0 [ set relevantDistance_m 0 ] ; no negative distances!
let detProb e ^ (Lambda_detectProb * relevantDistance_m)
if relevantDistance_m > ForagingRangeMax_m
  [ set detProb 0 ] ; patch is beyond the colonies foraging range
report precision detProb 10
end
```

, *****

to-report DetermineCaste_REP [whoCol]

```
_ let mycaste "worker" ; bee will develop into a worker, unless it becomes a queen
_ ; it will be a queen if individual weight and colony conditions for becoming a queen are both
fulfilled:
_ if (weight_mg >= [ dev_Q_DeterminationWeight_mg ] of OneSpecies speciesID ; individual criterion
_ and [ queenProduction? ] of colony whoCol = true) ; colony criterion
_ [ set mycaste "queen" ]
_ report mycaste
end
```

, *****

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

to Development_EggsProc

```
; ; procedure checks if eggs develop into larvae
set cumTimeEgg_d cumTimeEgg_d + 1
if cumIncubationReceived_kJ >= [ devIncubationHatchingTH_kJ ] of OneSpecies speciesID
  and broodAge >= [ devAgeHatchingMin_d ] of OneSpecies speciesID
  [
    set stage "larva"
    set color white
    if ploidy = 1 [ set color yellow ] ; male larvae are represented by yellow bars on the interface
    let memoNumber number
    ask colony colonyID [set totalLarvaeProduced totalLarvaeProduced + memoNumber]
  ]
end
```

```
; *****
```

to Development_LarvaeProc [whoCol]

```
; procedure checks if larvae develop into pupa
; Development factors larva: age, incubation and weight - option to develop into queen!
set cumTimeLarva_d cumTimeLarva_d + 1
if caste = "undefined" and cumTimeLarva_d = [ dev_larvalAge_QueenDetermination_d ] of
OneSpecies speciesID ; age of determination is independent of time spent as egg
[
  set caste DetermineCaste_REP whoCol ; this reporter-procedure determines the caste
  if caste = "queen" [ set color orange ]
  if caste = "undefined" [ AssertionProc "Assertion violated: undefined caste!
(Development_LarvaeProc)" ]
]

if caste = "worker" or caste = "male" ; larvae develop into pupae as soon as they 1) received
enough incubation, and 2) they are old enough and 3.) heavy enough
  and cumIncubationReceived_kJ >= [ devIncubationPupationTH_kJ ] of OneSpecies speciesID
  and broodAge >= [ devAgePupationMin_d ] of OneSpecies speciesID
  and weight_mg >= [ devWeightPupationMin_mg ] of OneSpecies speciesID
  [
    set stage "pupa"
    set color brown
    if ploidy = 1 [ set color grey - 2 ]
    let memoNumber number
    ask colony colonyID [set totalPupaeProduced totalPupaeProduced + memoNumber]
  ]

if caste = "queen" ;
  and cumIncubationReceived_kJ >= [ devIncubation_Q_PupationTH_kJ ] of OneSpecies speciesID
  and broodAge >= [ devAge_Q_PupationMin_d ] of OneSpecies speciesID
  and weight_mg >= [ devWeight_Q_PupationMin_mg ] of OneSpecies speciesID
  [
    set stage "pupa"
```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
    set color red
    let memoNumber number
    ask colony colonyID [set totalPupaeProduced totalPupaeProduced + memoNumber]
  ]

end

, *****
;
```

to Development_Mortality_AdultsProc

```
if caste = "worker" ; behavioural development workers
[
  if adultAge > [ maxLifespanWorkers ] of OneSpecies speciesID [ DieProc "Worker: adultAge >
maxLifespanWorkers" ]
]

if caste = "male" ; death of adult males after max lifespan
[
  if adultAge > MaxLifespanMales [ DieProc "Male: adultAge > MaxLifespanMales" ]
]

if caste = "queen" and mated? = true and colonyID >= 0 and [ allAdultWorkers ] of colony colonyID >
0 ; i.e. if colony is in the social phase
[
  set thForagingNectar ThresholdLevelREP "nectarForaging" "QueenSocialPhase"
  set thForagingPollen ThresholdLevelREP "pollenForaging" "QueenSocialPhase"
  set thNursing ThresholdLevelREP "nursing" "QueenSocialPhase"
  set thEggLaying ThresholdLevelREP "eggLaying" "QueenSocialPhase"
]

if MortalityAdultsBackground_daily > 0 and random-float 1 > MortalityAdultsBackground_daily ;
MortalityAdultsBackground_daily = 0 so no effect!
[ DieProc "Adult bee: mortality in colony" ]
end

, *****
;
```

to Development_PupaeProc

```
; procedure checks if pupae develop into adults
set cumulTimePupa_d cumulTimePupa_d + 1 ; potential output (time spent as pupa)
if caste = "worker"
[
  ; Development factors pupae: age & incubation:
  if cumulIncubationReceived_kJ >= [ devIncubationEmergingTH_kJ ] of OneSpecies speciesID
    and broodAge >= [ devAgeEmergingMin_d ] of OneSpecies speciesID
    [
      set stage "adult"
    ]
  ]
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
set brood? false
set color black
```

```
let newWorkers number ; saves the cohort size
set TotalAdultsEverProduced TotalAdultsEverProduced + number
ask colony colonyID [set totalAdultsProduced totalAdultsProduced + newWorkers]
ask colony colonyID [set totalWorkersProduced totalWorkersProduced + newWorkers]
```

```
; bee crop and pollen capacity based on weight using (1) linear formula (2) pollen:crop ratio
from HBs in BEEHAVE, with both
```

```
; having an upper limit set by species-own variables.
set cropvolume_myl CropAndPelletSizeREP "nectar"
set pollenPellets_g CropAndPelletSizeREP "pollen"
```

```
set thEgglaying ThresholdLevelREP "eggLaying" "worker"
set thForagingNectar ThresholdLevelREP "nectarForaging" "worker"
set thForagingPollen ThresholdLevelREP "pollenForaging" "worker"
set thNursing ThresholdLevelREP "nursing" "worker"
set glossaLength_mm ProboscisLengthREP
```

```
]
]
```

```
; NOTE: in bumblebees (B. terrestris) diploid males develop into (sterile) adults (Duchateau et al. 1994)
```

```
; (dipl. males can also mate but these queens are not able to establish a colony (Duchateau & Marien 1995) and are removed (in QueensLeavingNestProc)
```

```
if caste = "male" ; MALE PUPAE - might develop into adult males
```

```
[
  if cumullIncubationReceived_kj >= [ devIncubationEmergingTH_kj ] of OneSpecies speciesID
  and broodAge >= [ devAgeEmergingMin_d ] of OneSpecies speciesID
```

```
[
  set brood? false
  set stage "adult"
  set color green
  let newMales number
  set TotalAdultsEverProduced TotalAdultsEverProduced + number
  set TotalAdultMalesEverProduced TotalAdultMalesEverProduced + number
  ask colony colonyID
  [
    set totalAdultsProduced totalAdultsProduced + newMales
    set totalMalesProduced totalMalesProduced + newMales
  ]
]
```

```
]
]
```

```
if caste = "queen" ; QUEEN PUPAE - might develop into adult queens
```

```
[
  if cumullIncubationReceived_kj >= [ devIncubation_Q_EmergingTH_kj ] of OneSpecies speciesID
  and broodAge >= [ devAge_Q_EmergingMin_d ] of OneSpecies speciesID
```

```
[
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    let yearEndSeason (365 * ceiling (ticks / 365)) + [seasonStop] of OneSpecies speciesID ;
prevent bees from setting emergingDate past the end of season
    let yearStartSeason (365 * ceiling (ticks / 365))
    while [ emergingDate <= yearStartSeason OR emergingDate > yearEndSeason ] ; add start
season to the while statement
    [ set emergingDate (365 * ceiling (ticks / 365)) ; emerging from hibernation next year on day
"emergingDay_mean" (+- s.d.)
        + round random-normal [ emergingDay_mean ] of OneSpecies speciesID ; mean
            [ emergingDay_sd ] of OneSpecies speciesID ; SD
    set stage "adult"
    set brood? false
    set color red
    let newQueensProduced number
    if not member? colonyID QueensProducingColoniesList
        [ set QueensProducingColoniesList lput colonyID QueensProducingColoniesList ]
    set TotalAdultsEverProduced TotalAdultsEverProduced + number
    set TotalAdultQueensEverProduced TotalAdultQueensEverProduced + number
    ask colony colonyID
    [
        set totalQueensProduced totalQueensProduced + newQueensProduced
        set totalAdultsProduced totalAdultsProduced + newQueensProduced
    ]
    set cropvolume_myl CropAndPelletSizeREP "nectar"
    set pollenPellets_g CropAndPelletSizeREP "pollen"
    set thEgglaying ThresholdLevelREP "eggLaying" "youngQueen"
    set thForagingNectar ThresholdLevelREP "nectarForaging" "youngQueen"
    set thForagingPollen ThresholdLevelREP "pollenForaging" "youngQueen"
    set thNursing ThresholdLevelREP "nursing" "youngQueen"
    set glossaLength_mm ProboscisLengthREP
    ]
    ]

    if caste = "undefined" [ AssertionProc "Assertion violated: undefined caste!
(Development_PupaeProc)" ]
end

```

```

; *****
;

```

to DevelopmentProc

```

ask bees
[
    let whoCol colonyID
    ifelse stage = "adult"
    [
        set adultAge adultAge + 1
        if adultAge > 700 [ AssertionProc "Assertion violated: Bee with 2 hibernations!" ]
        if brood? = true [ AssertionProc "Assertion violated: adult bee with brood? = true
(DevelopmentProc)" ]
        Development_Mortality_AdultsProc
    ]
]

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

]
[
  set broodAge broodAge + 1
  if brood? = false [ AssertionProc "Assertion violated: brood with brood? = false
(DevelopmentProc)" ]
  if stage = "pupa" [ Development_PupaeProc ]
  if stage = "larva" [ Development_LarvaeProc whoCol ]
  if stage = "egg" [ Development_EggsProc ]
]
if xcor + StepWidth < max-pxcor and mated? = false ; move graphic representation of bees on GUI
[ set xcor xcor + StepWidth ]
]
end

, *****
,

```

to DieProc [causeOfDeath]

; calls the actual "die" command for all biologically relevant agents and keeps track of all their deaths

if breed = bees

```

[
  set TotalBeesEverDied TotalBeesEverDied + number
  if number < 1 [ show ticks AssertionProc "Less than 1 bee in bee agent (CheckNumbersProc)" ]

```

; Remove dying bees from the ActiveBeesSortedList:

if member? who ActiveBeesSortedList

```

[
  set ActiveBeesSortedList filter [? != who] ActiveBeesSortedList ]
die
]
```

if breed = colonies

```

[
if eusocialPhaseDate + switchPointDate + competitionPointDate < NotSetHigh
[
output type ticks id 1stWorker SP CP QPD death #Q #M: "
output type ticks output type " "
output type who output type " "
output type eusocialPhaseDate output type " "
output type switchPointDate output type " "
output type competitionPointDate output type " "
output type queenproductiondate output type " "
output type ticks output type " "
output type totalQueensProduced output type " "
output type totalMalesProduced output print " "
]

```

; instead of removing colony, change breed to deadCol

; also kill store bars and change agent into a small white dot on the 2D view

ask storebars with [storeColonyID = [who] of myself] [die]

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
set breed deadCols
set size 1
set label ""
set color white
set shape "circle"
set colonyDeathDay ticks
if not showDeadCols? [ ht ]
]
```

```
if breed = badgers
[
output-show causeOfDeath
die
]
```

```
if breed != deadCols [ AssertionProc "Zombie alarm in DieProc" ] ; only dead colonies are supposed
to survive DieProc
```

```
end
```

```
,,,, *****
```

to DrawCohortsProc

```
ask colonies
[
let whoCol who ; saves colony ID
if count bees with [ colonyID = whoCol and shape = "halfline" ] > 0
[
let currentAge 0
let maxAge 1 + [ broodAge + adultage ] of
max-one-of bees with [ colonyID = whoCol and shape = "halfline" ] [ broodAge + adultage ]
repeat maxAge
[
let cohortSize sum [number] of Bees
with [ broodAge + adultage = currentAge
and colonyID = whoCol
and shape = "halfline" ] ; i.e. not the mother queen
ask bees with [ broodAge + adultage = currentAge
and colonyID = whoCol
and shape = "halfline" ]
[ set size cohortSize * CohortSymbolSize ]
set currentAge currentAge + 1
]
]
]
end
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

, *****
;

to EgglayingProc

```

let mother caste ; to distinguish queen and worker laid eggs
let newCohorts 1 ; this will later be changed for IBM colonies
let beesInCohort [ batchsize ] of OneSpecies speciesID ; this will later be changed for IBM colonies
let eggWeight [devWeightEgg_mg] of OneSpecies speciesID
let pollenToMass [pollenToBodymassFactor] of OneSpecies speciesID

if [ cohortBased? ] of colony colonyID = false ; i.e. IBM colonies..
[
    set newCohorts [ batchsize ] of OneSpecies speciesID ; .. number of "cohorts" = batchSize, as
    each "cohort" contains only a single bee (as IBM colony)..
    set beesInCohort 1 ; .. with only 1 bee in each
]

; Pollen cost is total mass of laid eggs * the conversion of pollen to bee body mass
; Energy cost is amount needed by female to facilitate replacement lost pollen
let pollenCost_g beesInCohort * eggWeight * pollenToMass / 1000
let energyCost_kJ pollenCost_g * EnergyRequiredForPollenAssimilation_kJ_per_g

if [ pollenStore_g ] of colony colonyID > pollenCost_g and [ energyStore_kJ ] of colony colonyID >
energyCost_kJ
[ ; eggs can only be laid, if enough energy and pollen is present!
    hatch newCohorts ; "hatch" command, as "create" is not possible in a turtle context
    [
        ifelse mother = "queen"
        [ ; queens can produce male and female offspring:
            ifelse ticks > [ switchPointDate ] of colony colonyID ; after the switch point, only males are
            produced
                [ set ploidy 1 ] ; 1: haploid male
                [ set ploidy 2 ] ; 2: diploid bee (worker, queen or diploid male)
            ]
        [ ; workers can only produce male offspring:
            set ploidy 1 ; 1: haploid male
        ]
        EggsParameterSettingProc beesInCohort ; calls the procedure EggsParameterSettingProc and
        transfers the local variable beesInCohort
    ]

    ; Remove costs from store
    ask colony colonyID
    [
        set pollenStore_g pollenStore_g - pollenCost_g
        set energyStore_kJ energyStore_kJ - energyCost_kJ
    ]
]
set personalTime_s personalTime_s + 24 * 3600 ; will be reset on next morning!

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

end

, *****
,

to EggsParameterSettingProc [beesInCohort]

```
; sets parameter values for new eggs
; alleles of the egg:
let shiftDrawnCohorts 1 ; (1) to show cohorts above the colony
let myAllelesList []
set myAllelesList fput one-of allelesList myAllelesList ; egg gets (only) one allele from its mother
set allelesList myAllelesList
set caste "undefined" ; "undefined", "queen", "worker", "male"
ifelse ploidy = 1 ; haploid males
[
  set color violet
  set caste "male"
]
[ ; females and diploid males:
  if ploidy != 2 [ AssertionProc "Wrong ploidy! (EggsParameterSettingProc)" ]
  set color blue
  set allelesList fput one-of spermathecaList allelesList ; diploid bees get another allele from their
  father/spermatheca
  if SexLocus? = true ; if alleles refer to the sex locus..
    and item 0 allelesList = item 1 allelesList ; .. and bee is homozygous..
    [ set caste "male" ] ; .. it becomes a diploid male!
  ]
set spermathecaList [] ; eggs haven't mated yet..
set size CohortSymbolSize
set shape "halfline"
__set-line-thickness 0.5
set heading 0
set number beesInCohort
set TotalBeesEverProduced TotalBeesEverProduced + number
set activity "resting"
set adultAge 0 ; (changed from -1. To stop both brood and adultAge incrementing in tick of
maturity/emergence)
set brood? true
set broodAge 0 ; set to 0 as eggs are created with the "hatch" command
set cumTimeEgg_d 0 ; Set to 0, not mother's value
set cumTimeLarva_d 0 ; Set to 0, not mother's value
set cumTimePupa_d 0 ; Set to 0, not mother's value
set cropVolume_myl 0 ; now based on weight, has to be set on emergence
set pollenPellets_g 0 ; now based on weight, has to be set on emergence
set currentFoodsource -1 ; not set yet
set nectarsourceToGoTo -1 ; not set yet
set pollensourceToGoTo -1 ; not set yet
set stage "egg" ; egg, larva, pupa, adult
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

set mated? false

set thEgglaying ThresholdLevelREP "eggLaying" "egg"
set thForagingNectar ThresholdLevelREP "nectarForaging" "egg"
set thForagingPollen ThresholdLevelREP "pollenForaging" "egg"
set thNursing ThresholdLevelREP "nursing" "egg"

set activityList []

set knownMasterpatchesNectarList []

set knownMasterpatchesPollenList []

set weight_mg [devWeightEgg_mg] of OneSpecies speciesID

set cumullIncubationReceived_kJ 0

set emergingDate NotSetLow

set expectation_NectarTrip_s 0

set expectation_PollenTrip_s 0

set glossaLength_mm 0

set nectarLoadSquadron_kJ 0

set personalTime_s 0

set pollenForager? false

set pollenLoadSquadron_g 0

; location of egg cohort on the interface is relative to its colony's location:

- if [xcor] of Colony colonyID - [devAgeEmergingMin_d / 10] of OneSpecies speciesID > min-pxcor
__ set xcor [xcor] of Colony colonyID - [devAgeEmergingMin_d / 10] of OneSpecies speciesID]

- if [ycor] of Colony colonyID + shiftDrawnCohorts < max-pycor
__ set ycor [ycor] of Colony colonyID + shiftDrawnCohorts]

ifelse ShowCohorts? = false

[hide-turtle]

[show-turtle]

ask colony colonyID [set totalEggsProduced totalEggsProduced + beesInCohort]

end

to EmergenceNewQueensProc

; new queens emerge from hibernation and found new colonies (note: most queens will still be represented as cohorts here!)

; Winter survival (survivalProb) is calculated from Beekman et al 1998 (*Entomologia Experimentalis et Applicata* 89: 207–214, 1998)

; Fig. 1B: survival prob. is calculated from proportion of survivors to survivors + non-survivors. Fitted a sigmoid curve to the left site only, as the low surv. prob. of heavy queens

; is an artefact of the treatment:

; "One would expect that queens with the highest weight will survive diapause. It is therefore surprising that the initial weight distribution of dead queens exceeds that of the surviving queens (Figure 1B and 1C).

; However, in 1993 the average initial weight of the queens was highest and in this period the most severe diapause regimes (6 or 8 months) were started. Since the majority of the queens that were given a treatment

; with a length of 6 or 8 months died, the initial weight distribution of dead queens exceeds that of the surviving queens."

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
;cohorts split into individuals before being challenged for over-winter survival
```

```
ask bees with [emergingDate = ticks]
```

```
[
```

```
  ; EMERGING:
```

```
  if caste != "queen" [ show "WARNING! Something else than a queen emerged from hibernation!
```

```
(1)" ]
```

```
  set activity "emerging"
```

```
  ; thresholds are updated:
```

```
  set thEggLaying ThresholdLevelREP "eggLaying" "QueenInitiationPhase"
```

```
  set thForagingNectar ThresholdLevelREP "nectarForaging" "QueenInitiationPhase"
```

```
  set thForagingPollen ThresholdLevelREP "pollenForaging" "QueenInitiationPhase"
```

```
  set thNursing ThresholdLevelREP "nursing" "QueenInitiationPhase"
```

```
  ; HATCHING INDIVIDUALS:
```

```
  ; cohort based queens become individuals:
```

```
  let hatchlings number - 1 ; for cohort based queens: bee needs to be "cloned" cohortsize - 1 times!
```

```
  set number 1 ; new queens are individuals now (not cohorts)
```

```
  hatch hatchlings ; the "clones" of the originally cohort-based queenagent are created
```

```
]
```

```
ask bees with [activity = "emerging"]
```

```
[
```

```
  if caste != "queen" [ show "WARNING! Something else than a queen emerged from hibernation!
```

```
(2)" ]
```

```
  ; WINTER MORTALITY:
```

```
  ; Queen has a risk of dying over winter:
```

```
  if WinterMortality? = true and random-float 1 > WintermortalityProbREP
```

```
    [ DieProc "winter mortality"]
```

```
  ; AFTER SURVIVAL:
```

```
  set activity "resting"
```

```
  set colonyID -1 ; queens haven't found a nest site yet nor started a colony
```

```
  ifelse ShowQueens? = true
```

```
    [show-turtle]
```

```
    [hide-turtle]
```

```
]
```

```
; create new colonies for freshly emerged queens:
```

```
if count bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ] > 0 ; i.e.
```

```
(relatively) newly emerged queens (though might have emerged a few days ago)
```

```
[
```

```
  ; queens without a colony search for nest sites
```

```
  ask bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ]
```

```
[
```

```
  ifelse count [nestsiteFoodsourceList] of onespecies speciesID > 0 ;check that the habitat contains  
  suitable foodsources for this species to nest in
```

```
  [
```

```
    ifelse speciesName != "Psithyrus"
```

```
      [ NestSitesSearchingProc ] ; social BB
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
[ PsithyrusNestSearchProc ] ; cuckoo BB
if ShowSearchingQueens? = false [hide-turtle]
]
[
  DieProc (word "no suitable foodsources for nesting exist for " speciesname) ; kill off bees with
no chance of finding a nest site
]
]
; if successful, they build a new nest:
if count bees with [ activity = "nestConstruction" ] > 0
[ CreateColoniesProc ]
if count bees with [ activity = "nestConstruction" ] > 0
[ show "That's weird - no constructions here, please!" ]
]
end
```

```
, *****
,
```

to FeedLarvaeProc

; determines how much nectar and pollen is fed to larvae in each colony,
; calculates the resulting weight gain of the larvae and updates the colony stores

ask colonies

[

let myColony who

; RELATIVE AMOUNTS TO BE FED:

; Calculate amount of NP to be fed to larvae relative to the amount they require for maximum
growth

let relativePollenToBeFed 0 ; may be updated below

let relativeEnergyToBeFed 0

; This will be set based on how large the stores are relative to the ideal stores (these have already
been filled through foraging today)

; Both stores must be greater than 0, as pollen and nectar are both required for growth

if idealPollenStore_g * idealEnergyStore_kj > 0 ; i.e. if both > 0

[

set relativePollenToBeFed pollenStore_g / idealPollenStore_g

set relativeEnergyToBeFed energyStore_kj / idealEnergyStore_kj

; set values to be bound by 0 1. Added bound by 0 because values can be lower if the energyStore
is negative (this is okay, because

; the colony will die at the start of the next tick). Negative values lead to energy being taken from
the larvae and added back to the store.

set relativePollenToBeFed median (list 0 1 relativePollenToBeFed) ; input of the median command
needs to be a list..

set relativeEnergyToBeFed median (list 0 1 relativeEnergyToBeFed) ; ..picks the "middle" of the
three values, i.e. usually relativePollen/EnergyToBeFed

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

]

```
; So set the growth limiting factor as the lowest of either relativeEnergy or relativePollen
; based on Liebig's law of the minimum, larval growth is assumed to be limited by only one factor:
let growthLimitingFactor relativePollenToBeFed
if relativeEnergyToBeFed < relativePollenToBeFed ; amount of nectar fed is adjusted to the amount
of pollen fed
[ set growthLimitingFactor relativeEnergyToBeFed ]
```

```
; ACTUAL FEEDING OF EACH INDIVIDUAL LARVA:
let totalPollenFedToday_g 0 ; sums up the total amount of pollen a colony feeds to the larvae
ask bees with [ stage = "larva" and colonyID = myColony ]
[
; Calculate pollen gained based on conversion to max weight gain adjusted by limiting factor
; amount of pollen fed to a single larva (no "number" here as it refers to amount an individual
larva gets)
let pollenReceivedToday_mg growthLimitingFactor * ((MaxWeightGainToday_mg_REP who) /
([pollenToBodymassFactor] of OneSpecies speciesID))
if pollenReceivedToday_mg > [ pollenStore_g ] of colony myColony * 1000 ; to avoid negative
pollen stores
[
set pollenReceivedToday_mg [ pollenStore_g ] of colony myColony * 1000
if pollenReceivedToday_mg < 0 [ set pollenReceivedToday_mg 0 ] ; in case of negative pollen
stores, larvae are not fed at all!
]
```

```
; Update the larva's weight
let oldWeight_mg weight_mg
set weight_mg weight_mg + pollenReceivedToday_mg * [pollenToBodymassFactor] of OneSpecies
speciesID
if weight_mg < 0 [ AssertionProc "BUG in FeedPOLLENProc" ]
if weight_mg < oldWeight_mg [ AssertionProc "BUG in FeedPOLLENProc: WeightLoss" ]
; Update the total pollen to be taken from the store by the number of individuals in the cohort
set totalPollenFedToday_g totalPollenFedToday_g + (number * pollenReceivedToday_mg / 1000)
; multiplied by "number" here as it refers to the total costs for the colony
]
```

```
; Update the total energy required to assimilate the pollen consumed
let totalEnergyFedToday_kJ totalPollenFedToday_g *
EnergyRequiredForPollenAssimilation_kJ_per_g
```

```
; REMOVING RESOURCES FROM THE STORES:
set pollenStore_g pollenStore_g - totalPollenFedToday_g
if pollenStore_g < 0 [ type "negative pollen store! Ticks: " show ticks ]
set energyStore_kJ energyStore_kJ - totalEnergyFedToday_kJ ; negative energy store doesn't
matter as it results in the death of the colony the next morning (in UpdateColoniesProc)
]
end
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
... *****  
,,,
```

to FoodsourcesInFlowerAndRangeProc

; called by a colony; creates 5 lists, containing the who of foodsources or masterpatches offering nectar, offering pollen or offering either nectar or pollen within the foraging range

```
set pollenInFlowerAndRangeList []  
set nectarInFlowerAndRangeList []  
set allSourcesInFlowerAndRangeList []  
let pol []  
let polM []  
let nec []  
let necM []
```

```
foreach allPatchesInRangeList ; all patches within foraging range are addressed  
[  
  ask foodsource ?  
  [  
    if pollen_g > 0 ; if they contain pollen..  
    [  
      set pol lput who pol ; .. their ID is added to the list pol  
      set polM lput masterpatchID polM ; .. and their masterpatch is added to the list polM  
    ]  
    if nectar_myl > 0 ; similar for nectar  
    [  
      set nec lput who nec  
      set necM lput masterpatchID necM  
    ]  
  ]  
]
```

```
set polM remove-duplicates polM ; make sure, a masterpatch occurs only once in the polM list  
set necM remove-duplicates necM ; ditto for necM  
set pollenInFlowerAndRangeList pol  
set masterpatchesWithPollenlayersInFlowerAndRangeList polM  
set nectarInFlowerAndRangeList nec  
set masterpatchesWithNectarlayersInFlowerAndRangeList necM  
set allSourcesInFlowerAndRangeList remove-duplicates (sentence pollenInFlowerAndRangeList  
nectarInFlowerAndRangeList) ; combines the nectar and pollen list into a single list
```

```
end
```

```
, *****  
;
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

to-report Foraging_bestLayerREP [myCurrentFoodsource]

; reports most profitable foodsource ("layer") within the currently used flower patch, based on minimal handling time

```
let memoBestHandlingtime notSetHigh
let memoBestEEF notSetLow
let memoBestPatch -1
let myBeeID who
let distanceColonyFoodpatch_m 0 ;; distance (same for all layers!) will be set now:
ask colony colonyID
[
  set distanceColonyFoodpatch_m ; distance between the colony and the food patch
  distance (Foodsource myCurrentFoodsource) ; the distance in NetLogo patches
  / SCALING_NLpatches/m ; div. by scaling => distance in m
]

foreach [ layersInPatchList ] of foodsource myCurrentFoodsource ; for each foodsource of the bees
'layergroup', the handling time is calculated
[
  let currentLayer ?
  if pollenforager? = true and [ pollen_g ] of foodsource currentLayer > 0 ; only patches that
  actually provide pollen are considered
  [
    ask bee myBeeID
    [
      ; handling time is determined:
      let handlingTime_s HandlingTime_s_REP currentLayer pollenforager?
      if handlingTime_s < memoBestHandlingtime ; and if it is the shortest so far..
      [
        set memoBestPatch currentLayer ; the ID of this foodsource..
        set memoBestHandlingtime handlingTime_s ; and the handling time are saved
      ]
    ]
  ]

  if pollenforager? = false and [ nectar_myl ] of foodsource currentLayer > 0 ; only patches that
  actually provide nectar are considered
  [
    ask bee myBeeID
    [
      ; handling time is determined:
      let handlingTime_s HandlingTime_s_REP currentLayer pollenforager?
      let energyCostsThisLayer_kJ ; energy needed to exploit this layer:
      (
        2 * distanceColonyFoodpatch_m ; bees fly to and return from food patch
        ; plus distances they fly within the patch, but reduced by rests on flowers:
        + handlingTime_s
        * [ flightVelocity_m/s ] of OneSpecies speciesID ; [s] * [m/s] = [m]
        * EnergyFactorOnFlower
      )
    ]
  ]
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWART: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```

    * [ flightCosts_kJ/m/mg ] of OneSpecies speciesID * weight_mg ; flight costs (kJ) per m,
    dependent on the bees' weight
    * number ; [m] * [kJ/m/mg] * [mg] => [kJ]
    ; energy gained when exploiting this layer:
    let energyGainThisLayer_kJ cropvolume_myl
        * EnergySucrose_kJ/mymol ; [ul] * [kJ/umol] => [kJ/mol * l]
        * [ nectarConcentration_mol/l ] of Foodsource currentLayer ; [kJ/mol * l] * [mol/l]
=> [kJ]
    ; energetic efficiency of exploiting this layer:
    let eef (energyGainThisLayer_kJ - energyCostsThisLayer_kJ) / energyCostsThisLayer_kJ

    if eef > memoBestEEF ; if it is the energetically best so far...
    [
        set memoBestPatch currentLayer ; the ID of this foodsource..
        set memoBestEEF eef ; and the energetic efficiency are saved
    ]
]
]
]
report memoBestPatch ; this might be negative, if no foodsource was found!
end

```

; *****

to Foraging_collectNectarPollenProc

```

; foragers with activity = "collectPollen" or activity = "expForagingP" OR activity = "collectNectar" or
activity = "expForagingN" can gather food from a food source. No other bees are addressed
if activity = "collectPollen" or activity = "expForagingP"
[
    ; amount of pollen that can be collected by the forager squadron
    ; does patch still have any pollen?:
    ifelse [ pollen_g ] of Foodsource currentFoodsource > 0
    [ ; the forager will then be bringing pollen:
        set pollenLoadSquadron_g min list ; takes the smaller value of an ad hoc created list with two
items: 1st: max. pollen bee (cohort) can carry, 2nd: pollen left at patch.
        (pollenPellets_g * number) ; 1st item: max. pollen a bee (cohort) can carry
        ([ pollen_g ] of Foodsource currentFoodsource) ; 2nd item: max. pollen bee (cohort) can carry
        set activity "bringingPollen"
        set activityList lput "P" activityList
        ; amount of pollen at the patch is reduced
        let memoNumber number
        let memoPollenLoad pollenLoadSquadron_g
        ask foodsource currentFoodsource
        [
            set pollen_g pollen_g - memoPollenLoad
            set cumulPollenVisits cumulPollenVisits + memoNumber ; all pollen visits at patch, ever
        ]
    ]
]
[

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```

    set activity "returningEmpty"
    set activityList lput "Ep" activityList
  ]
]

if activity = "collectNectar" or activity = "expForagingN"
[
; does patch still have any nectar?:
ifelse [ nectar_myl ] of Foodsource currentFoodsource > 0
[ ; the forager will then be bringing nectar:
  let nectarRemoved min list ; takes the smaller value of an ad hoc created list
    ; with two items: 1st: max. nectar a bee (cohort) can carry, 2nd: nectar left at
patch.
    (cropvolume_myl * number) ; 1st item: max. nectar a bee (cohort) can carry
    ([ nectar_myl ] of Foodsource currentFoodsource) ; 2nd item: max. pollen bee (cohort) can
carry

    set nectarLoadSquadron_kJ nectarRemoved * EnergySucrose_kJ/mymol * [
nectarConcentration_mol/l ] of Foodsource currentFoodsource
    ; set the nectar energy load with the amount removed

    set activity "bringingNectar"
    set activityList lput "N" activityList
    ; amount of nectar at the patch is reduced:
    let memoNumber number
    ask foodsource currentFoodsource
    [
      set nectar_myl nectar_myl - NectarRemoved
      set cumulNectarVisits cumulNectarVisits + memoNumber ; all nectar visits at patch, ever
    ]
  ]
[
; if NECTAR foodsource is EMPTY:
set activity "returningEmpty"
set activityList lput "En" activityList ; "Empty nectar"
]
]
end

; *****

```

to Foraging_costs&choiceProc

; costs in time, energy and mortality
; first bees with activity = "returningEmpty" and then bees with activity = "bringingNectar" or
activity = "bringingPollen" are addressed to calculate time and energy spent on the trip;
; finally Foraging_PatchChoiceProc is called, as the future patch/flowerspecies choice is based on the
time spent on the trip.

let persTimeSave personalTime_s ; the current time

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
let saveNumber number ; number of individuals, this cohort/forager agent represents
let saveWeight_mg weight_mg ; save variable for weight of individual/s
let saveFlightCosts_kJ/m/mg [ flightCosts_kJ/m/mg ] of OneSpecies speciesID ;FlightCosts set as
the same for all foragers
```

```
; Test for errors
if not member? caste ["worker" "queen"] [ AssertionProc (word stage " " caste " " who " shouldn't
be foraging (Foraging_costs&timeProc"))]
if saveFlightCosts_kJ/m/mg < 0 [ AssertionProc "saveFlightCosts local variable not set
(Foraging_costs&timeProc)"]
```

; EMPTY BEES:

```
if activity = "returningEmpty"
[
; nectar store in the colony is reduced to reflect the energy consumed during the trip:
let tripDuration_s 0
ask Colony colonyID
[
set energyStore_kJ energyStore_kJ - ( [ searchLength_m ] of OneSpecies speciesIDcolony *
saveFlightCosts_kJ/m/mg * saveNumber * saveWeight_mg )
set tripDuration_s [ searchLength_m ] of OneSpecies speciesIDcolony / [ flightVelocity_m/s ] of
OneSpecies speciesIDcolony
]
set personalTime_s personalTime_s + tripDuration_s ; some time has passed..
; a Bee dies during the foraging trip, unless she survives every single second:
let survivalChance (1 - MortalityForager_per_s * ForagingMortalityFactor) ; probability to survive a
single second of the foraging trip
^ tripDuration_s ; ... to survive EACH second of the trip
let mortalityRisk 1 - survivalChance ; risk to die = 1 - probability to survive
if random-float 1 < mortalityRisk [ DieProc "foraging: empty" ] ; does bee die?
]
```

; SUCCESSFUL FORAGERS:

; energy consumption of successful foragers:

```
let handlingTime_s 0
if activity = "bringingNectar" or activity = "bringingPollen"
[
if activity = "bringingNectar"
[
set handlingTime_s HandlingTime_s_REP currentFoodsource pollenForager?
set activity "expForagingN"
set activityList lput "Xn" activityList
]
if activity = "bringingPollen"
[
set handlingTime_s HandlingTime_s_REP currentFoodsource pollenForager?
set activity "expForagingP"
set activityList lput "Xp" activityList
]
let memoPatch currentFoodsource
let tripDuration_s 0
```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
ask Colony colonyID
[
  set energyStore_kJ energyStore_kJ -
  (
    2 * distance (Foodsource memoPatch) / SCALING_NLpatches/m ; bees have to fly to the
    patch and back, distance [NLpatches] / Scaling = [m]
    ; plus distances they fly within the patch, but reduced by rests on flowers
    + handlingTime_s
    * [ flightVelocity_m/s ] of OneSpecies speciesIDcolony * EnergyFactorOnFlower ; [kJ] = [m
    * kJ/m + kJ/m * s * m/s]
  )
  * saveFlightCosts_kJ/m/mg * saveWeight_mg ; flight costs (kJ) per m, dependent on the bees'
  weight
  * saveNumber ; multiplied by number of bees in the cohort
  set tripDuration_s (2 * distance (Foodsource memoPatch) / SCALING_NLpatches/m / [
  flightVelocity_m/s ] of OneSpecies speciesIDcolony) + handlingTime_s
]
set personalTime_s personalTime_s + tripDuration_s ; some time has passed..
let survivalChance (1 - MortalityForager_per_s * ForagingMortalityFactor) ; probability to survive a
single second of the foraging trip
; ForagingMortalityFactor (set on GUI): allows to easliy modify foraging
mortality
^ tripDuration_s ; ... to survive EACH second of the trip
let mortalityRisk 1 - survivalChance ; risk to die = 1 - probability to survive
if random-float 1 < mortalityRisk [ DieProc "foraging: N or P forager" ]
set activityList lput (word "HT:" precision handlingTime_s 1) activityList
]

if personalTime_s - persTimeSave <= 0 [ AssertionProc "No time - or negative time - passed for this
bee! (Foraging_costs&timeProc)" ]
Foraging_PatchChoiceProc personalTime_s - persTimeSave ; the bee makes a choice about where
to forage next, based on the trip duration
end

, *****
```

to-report Foraging_periodREP

```
let foragingPeriod_s -1
let foragingHoursList [ ]
let foragingHoursExample [ 0 3.1 0 0 0 1.5 0 0.1 0 0 1.7 1.6 0 0 0 0 0 0 0 1.5 5 0 3.2 0
0 0 0.2 0 0 0 0.1 0.9 5.9 3.5 6.9 1.3 7.7 2.3 4.6 2.2 0.5 9.2 0 8 3.2 4.1 0 9 9.1 7.3 5.7 4.9
0 12.1 6.5 7.9 7.9 11.1 2.8 0 2.8 6 5.7 0 4 10.1 2.9 10.1 0 11.4 6.3 9.9 4.4 7.5 8 12.3 8.7
10.3 3.7 11.3 13.2 14 4.2 7.7 8.2 7.2 9.2 5 13.1 10.5 3.5 11.1 13.6 6.2 8.4 7.8 8.5 9.8 6.5
4.1 10.8 12.5 15.1 10.1 4.3 7 9.4 8.9 7.5 7.8 6.6 11.4 12.1 12.4 11.9 10.1 14.7 7.8 13.1 3.3
16.6 14.8 17.9 5.7 0.2 2.9 10 14.7 16.2 15.8 5.3 5.8 2.5 6 15.2 1.3 13.1 11.2 2 12.9 9.7 2.1
17.3 5.7 8.5 13.1 18.5 1.7 6.7 13.8 0.5 0.8 15.7 4.9 11.4 11.9 3.8 11.7 7.1 21.2 17.7 1.8
12.3 15.7 16.9 16.8 9.9 3.6 20.4 13 5.1 0.6 11.7 2.1 4.7 13.9 13.8 1.4 0.3 18.4 14.8 12.8 3.7
13.5 4.7 0.3 5.5 4 17.5 1.7 0.3 14.9 12.4 11.6 8.5 4.5 11.1 16 13.2 13.8 0.7 7.1 14.3 3.4 2.2
5.6 10.6 3.4 15.5 15.6 12.8 15 14 5.9 15.5 9.1 2 1 3.2 9.3 3 3.1 14 10.2 1 9.7 8.8 3.8 1.9
11.9 9.3 6.5 6.6 8.4 4.3 7.2 1.5 11.4 10.4 13.5 1.2 6 4.4 13.5 12.4 8 9.3 5.9 0.9 6.8 5.9 9.1
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
10.5 6 7.9 2.3 0.8 0 7.9 11 1.3 8.7 6.5 6.6 7.6 0 0 9.4 7.1 6.4 4 6.6 0 2.7 0 0 7.8 0 8.7
0.3 2 4.8 1.8 0.9 0 0 7.2 5.8 6.5 0 1.1 0 0 0.7 6.3 1.3 0 5.5 1.4 2.8 0 0 0 4 0 1.4 5.1 0 0
2.1 0 0.5 0 1 0 0 2.3 0 0 0 1.4 0.6 0 0 0 0 0.8 0 0 1 0.9 0 0 0 0 0 0 2.3 0 0 1.9 1.4 0 0
0 1.5 0 0 0 1 1.9 0 0 3.4 0 0 1 0 0 0 0 0 0 0 1.6]
```

```
if Weather = "foragingHoursExample"
[
  set foragingHoursList foragingHoursExample
  set foragingPeriod_s (item (day - 1) foragingHoursList) * 3600
]
if Weather = "Constant 24 hrs" [ set foragingPeriod_s 24 * 3600 ]
if Weather = "Constant 20 hrs" [ set foragingPeriod_s 20 * 3600 ]
if Weather = "Constant 16 hrs" [ set foragingPeriod_s 16 * 3600 ]
if Weather = "Constant 12 hrs" [ set foragingPeriod_s 12 * 3600 ]
if Weather = "Constant 11 hrs" [ set foragingPeriod_s 11 * 3600 ]
if Weather = "Constant 10 hrs" [ set foragingPeriod_s 10 * 3600 ]
if Weather = "Constant 9 hrs" [ set foragingPeriod_s 9 * 3600 ]
if Weather = "Constant 8 hrs" [ set foragingPeriod_s 8 * 3600 ]
if Weather = "Constant 7 hrs" [ set foragingPeriod_s 7 * 3600 ]
if Weather = "Constant 6 hrs" [ set foragingPeriod_s 6 * 3600 ]
if Weather = "Constant 5 hrs" [ set foragingPeriod_s 5 * 3600 ]
if Weather = "Constant 4 hrs" [ set foragingPeriod_s 4 * 3600 ]
if Weather = "Constant 3 hrs" [ set foragingPeriod_s 3 * 3600 ]
if Weather = "Constant 2 hrs" [ set foragingPeriod_s 2 * 3600 ]
if Weather = "Constant 1 hrs" [ set foragingPeriod_s 1 * 3600 ]
report foragingPeriod_s
end
```

```
, *****
```

to Foraging_searchingProc

; foragers with activity = "searching" may find a food source, other foragers (activity: collect nectar or pollen) don't do anything here

```
if activity = "searching"
[
  let chosenMasterpatch -1 ; bee hasn't found a food source yet
  let myMasterpatchesWithFoodList []
  let xcol NotSetHigh ; saves the location of the bee's colony to determine the detection probability
  let ycol NotSetHigh
  let pollenFor false
  if pollenForager? = true [ set pollenFor true ]
  ask colony colonyID
  [
    set xcol xcor
    set ycol ycor

    ifelse pollenFor = true
      [ set myMasterpatchesWithFoodList masterpatchesWithPollenlayersInFlowerAndRangeList ] ;
    only masterpatches are considered, otherwise, detection prob. would increase with the number of
    flowerspecies at a patch!
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
[ set myMasterpatchesWithFoodList masterpatchesWithNectarlayersInFlowerAndRangeList ]
]

foreach shuffle myMasterpatchesWithFoodList ; shuffled only once, not every time a new item is
addressed!
[
  if random-float 1 < DetectionProbREP ? xcol ycol ; all items in list are addressed, hence
  chosenMasterpatch may be set several times - only last patch detected is the patch chosen!
  [ set chosenMasterpatch ? ] ; this is a masterpatch that has at least 1 layer currently
  providing the forage the bee is searching for
]

ifelse chosenMasterpatch >= 0 ; if the bee has found a patch:
[
  set currentFoodsource Foraging_bestLayerREP chosenMasterpatch ; the bees new food source is
  then the best layer at that patch (based on handling time) (only sources actually providing the food
  the bee is after are considered)
  ifelse currentFoodsource >= 0
  [
    ifelse pollenForager? = true
    [
      set knownMasterpatchesPollenList fput chosenMasterpatch knownMasterpatchesPollenList ;
      food source is added to the list of known pollen patches
      set knownMasterpatchesPollenList Foraging_SortKnownPatchesListREP
      knownMasterpatchesPollenList ; the list is now sorted again by distances, with duplicates being
      removed
      set activity "collectPollen"
      set activityList lput "cP" activityList
    ]
    [
      set knownMasterpatchesNectarList fput chosenMasterpatch knownMasterpatchesNectarList ;
      food source is added to the bees' list of known nectar patches
      set knownMasterpatchesNectarList Foraging_SortKnownPatchesListREP
      knownMasterpatchesNectarList ; the bees' list is now sorted again by distance, with duplicates being
      removed
      set activity "collectNectar"
      set activityList lput "cN" activityList
    ]
  ]
  [
    ; bee found a patch but with 0 nectar or pollen (because this foodsource was visited by bees and
    depleted earlier today)
    set activity "returningEmpty"
    set activityList lput "rE0" activityList
  ]
]
; otherwise, if bee does not find a patch:
set activity "returningEmpty"
set activityList lput "rE" activityList
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
]
if activity = "searching" [ AssertionProc "Bee is still searching! (Foraging_searchingProc)" ]
end

, *****
;
```

to-report Foraging_SortKnownPatchesListREP [knownPatchesList]

```
; removes duplicates in list and sorts it by distances of masterpatches and the bee's colony
let newList []
set knownPatchesList remove-duplicates knownPatchesList ; duplicates are removed from the list
ask Colony colonyID ; this is the colony of the scouting bee
[ set newList sort-by [distance (Foodsource ?1) < distance (Foodsource ?2)] knownPatchesList ] ;
division by SCALING_NLpatches/m is not necessary here!
report newList
end
```

```
, *****
;
```

to Foraging_unloadingProc

```
; successful foragers (irrespective whether they are happy or not) unload their nectar or pollen load
ifelse activity = "expForagingN" or activity = "expForagingP" or activity = "returningUnhappyN" or
activity = "returningUnhappyP"
```

```
[
let nectarIncrease nectarLoadSquadron_kJ
let pollenIncrease pollenLoadSquadron_g

ask Colony colonyID ; load is added to the colony's stores:
[
set energyStore_kJ energyStore_kJ + nectarIncrease
set pollenStore_g pollenStore_g + pollenIncrease
]
ifelse activity = "expForagingN" or activity = "returningUnhappyN"
[ set activityList lput (word "N+" precision nectarIncrease 2) activityList ]
[ set activityList lput (word "P+" precision pollenIncrease 4) activityList ]
set nectarLoadSquadron_kJ 0
set pollenLoadSquadron_g 0
set personalTime_s personalTime_s + [ timeUnloading ] of OneSpecies speciesID
]
[ ; make sure bees with other activities don't carry nectar or pollen:
if nectarLoadSquadron_kJ + pollenLoadSquadron_g > 0
[ AssertionProc "Bee did not unload nectar or pollen in Foraging_unloadingProc!" ]
]
end
```

```
, *****
;
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

to ForagingProc

```
set TotalForagingTripsToday TotalForagingTripsToday + number
ifelse activity = "pollenForaging" ; if bee decided to collect pollen..
[
  set pollenForager? true      ; .. it becomes a pollen forager..
  set currentFoodsource pollensourceToGoTo
]
[
  set pollenForager? false    ; or otherwise a nectar forager
  set currentFoodsource nectarsourceToGoTo
]

ifelse currentFoodsource < 0 ; i.e. currentFoodsource does not refer to an existing food source
[
  set activity "searching"
  set activityList lput "S" activityList
]
[
  ifelse pollenForager? = true
  [
    set activity "collectPollen"
    set activityList lput "cP" activityList
  ]
  [
    set activity "collectNectar"
    set activityList lput "cN" activityList
  ]
]

Foraging_searchingProc ; unexperienced foragers search new flower patch
set activityList lput (word "(" currentFoodsource ")") activityList ; add foodsource to activityList
after search
Foraging_collectNectarPollenProc ; succesful scouts and experienced Foragers gather nectar
Foraging_costs&choiceProc ; energy costs for flights and trip duration
Foraging_unloadingProc ; ..and unload their crop & increase colony's honey store

if (pollenLoadSquadron_g + nectarLoadSquadron_kJ) > 0
  [ AssertionProc "Bee carries pollen or nectar after unloading! (ForagingProc)" ]
end
```

, *****

to Go

```
if AssertionViolated = true
[
  ask patches [ set pcolor red ]
  user-message "Assertion violated!"
  stop
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

; make sure, InspectTurtle refers to a colony:

if (count bees with [colonyID = InspectTurtle] = 0 and count Colonies > 0)

[set InspectTurtle [who] of one-of colonies]

if any? turtles with [who = InspectTurtle] and count Colonies > 0

[

if ([breed] of turtle InspectTurtle != Colonies)

[set InspectTurtle [who] of one-of colonies]

]

let continueWorking true

tick

UpdateMorning_Proc

NeedNectarPollenLarvaeTodayProc ; this calculates how much nectar and pollen is required for the brood

while [continueWorking = true ; still some time left today to do some work..

and count bees with [(caste = "worker" or caste = "queen") and (activity != "hibernate") and stage = "adult"] > 0 ; there are actually (active) bees, that can work

and count colonies > 0]

[

set ActiveBee NextActiveBeeREP ; runs faster than using "ask min-one-of bees with [(caste = "worker" or caste = "queen") and (activity != "hibernate") and stage = "adult"][personalTime_s]"

ask bee ActiveBee

[

ifelse personalTime_s > CallItaDay_s

[set continueWorking false]

[

set Daytime_s personalTime_s ; day time based on personal time of current bee

ifelse (floor (remainder personalTime_s 3600) / 60) >= 10 ; adds current personal time to activityList (hh:mm)

[set activityList lput (word floor (personalTime_s / 3600) ":" floor ((remainder personalTime_s 3600) / 60)) activityList]

[set activityList lput (word floor (personalTime_s / 3600) ":0" floor ((remainder personalTime_s 3600) / 60)) activityList]

ActivityProc

]

if colonyID = -1 [set personalTime_s CallItaDay_s + 1] ; if queen hasn't founded a colony yet, it won't be active for the rest of the day

]

]

ask bees with [stage = "adult" and activity != "hibernate" and caste != "male"]

[

set activity "resting"

set activityList lput "End" activityList

]

QueensLeavingNestProc ; young queens leave the nest to mate & hibernate

FeedLarvaeProc

QueenProductionDateProc

DevelopmentProc

MortalityBroodProc

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

BadgersOnTheProwlProc

OutputDailyProc

if ShowCohorts? = true [DrawCohortsProc]

if count Colonies + count Bees = 0 ~~and~~

└

└ if TimeToExtinction <= 0 [set TimeToExtinction ticks]

└ set Survived? 0

└ if StopExtinct? = true [stop]

└

end

to-report HandlingTime_s_REP [myPatch pollenPatch?]

; called by bee

; calculates the time [s] to gather of full load of nectar or pollen

; for nectar: based on Harder 1983: *Oecologia* 57:274-280

let maxHandlingTime_s 60 * 60 ; approx. max. from Ings et al. 2006, Fig. 1; *Journal of Applied Ecology*, 43,940–948; also comparable to data from Fig. 6 in Stelzer et al 2010, *PloS One*, 5(3), e9559

let handlingTime_s -999

let fillingLevel 0 ; minFillingLevel ; amount of food (nectar or pollen) currently at the patch relative to its max. value for today, calculated below

ifelse pollenPatch? = false

; NECTAR FORAGING:

[

ask foodsource myPatch

[

if nectarMax_myl > 0

[set fillingLevel nectar_myl / nectarMax_myl]

]

; Harder 1983, Fig. 4:

let W_beeWeight_g weight_mg / 1000

let G_lengthGlossa_mm glossaLength_mm ; mm

let C_CorollaDepth_mm [corollaDepth_mm] of Foodsource myPatch

let V_nectarVolume_myl [nectarFlowerVolume_myl] of Foodsource myPatch

let Ta_accessTime_s 0.3 + 0.04 * C_CorollaDepth_mm ; time to access a flower

let numerator log (V_nectarVolume_myl + 1) 10

let num 0.3 * W_beeWeight_g ^ 0.3333 * G_lengthGlossa_mm

let base (1.41 - C_CorollaDepth_mm / G_lengthGlossa_mm)

if base < 0.001 [set base 0.001] ; as 0 ^ -0.4 is not valid (in calculation of local variable den, see below)

let den (base ^ -0.4) - 0.3 * Ta_accessTime_s

let denominator log (num / den + 1) 10

let Ti_ingestionTime_s numerator / denominator

let handlingTimePerFlower_s maxHandlingTime_s ; handling time set to maximal value..

if fillingLevel > 0 ; avoid division by 0 ; .. unless there is nectar available, then it is recalculated

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

; (if the new value is larger than maxHandlingTime_s, it will be set back to
maxHandlingTime_s at the end of this procedure)
[
  set handlingTimePerFlower_s (
    (
      [ interFlowerTime_s ] of Foodsource myPatch ; the time to travel to the next
flower
      + Ta_accessTime_s ; + the time to test whether it contains
nectar
    )
    / fillingLevel ; divided by the filling level to account for depletion of the
patch
  )
  + Ti_ingestionTime_s ; + time to actually load the nectar, once a
filled flower is found
]
let flowersVisited 1 ; at least one flower has to be visited..
if V_nectarVolume_myl < cropvolume_myl [ set flowersVisited (cropvolume_myl /
V_nectarVolume_myl) ] ; but usually more than one flower is needed
set handlingTime_s handlingTimePerFlower_s * flowersVisited ; the time to find a flower and
empty it is then multiplied by the number of flowers, needed to fill the crop

] ; end if nectar forager

; POLLEN FORAGING:
[

  let timeInFlowers_s 257.4 ; time bee spends in flower(s) to collect 1 pollen load, derived (for poppy
flowers) from Raine & Chittka 2007, Tab. 1, "Number of flowers visited" times "Mean flower
handling time/ s" (mean of all three bouts)
  let flowersNeededForPollenLoad 58 ; Raine & Chittka 2007, Tab. 1 "Number of flowers visited"
(mean of all 3 bouts)
  ask foodsource myPatch ; get the filling level for this foodsource:
  [
    ifelse pollenMax_g > 0
    [ set fillingLevel pollen_g / pollenMax_g ]
    [ set fillingLevel 0 ] ; (this should actually never be the case)
  ]

  ifelse fillingLevel > 0
  [
    set handlingTime_s [ interFlowerTime_s ] of Foodsource myPatch ; the time to travel to the next
flower
    * flowersNeededForPollenLoad ; times the number of flowers needed to be
visited
    / fillingLevel ; divided by the filling level to account for depletion of the patch
    + timeInFlowers_s ; + time to actually collect the pollen, once a flower
with pollen is found
  ]
  [ set handlingTime_s maxHandlingTime_s ]
] ; end: if pollen forager

```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
if handlingTime_s > maxHandlingTime_s [ set handlingTime_s maxHandlingTime_s ]
report handlingTime_s
end
```

```
, *****
;
```

to-report MaxWeightGainToday_mg_REP [myID]

; calculates a larva's maximal weight gain during 24 hrs

```
let maxWeightGain_mg 0
```

```
let memoQPupationMax [ devWeight_Q_PupationMax_mg ] of OneSpecies speciesID ; max. weight
a queen pupa of this species can have
```

```
let memoWPupationMax [ devWeightPupationMax_mg ] of OneSpecies speciesID ; max. weight a
worker pupa of this species can have
```

```
ask bee myID
```

```
[
let myGrowthFactor [ growthFactor ] of OneSpecies speciesID ; growth factor depends on the
species
```

```
set maxWeightGain_mg (weight_mg * myGrowthFactor) - weight_mg ; i.e. a larva's max. weight
gain today
```

```
; weight can't exceed a caste specific maximal weight, i.e. no weight gain in this case!
```

```
; if the maximum new weight is greater than the pupation max, reduce maxWeightGain to
difference between pupation max and current weight
```

```
if caste = "queen" and weight_mg + maxWeightGain_mg > memoQPupationMax
```

```
[ set maxWeightGain_mg memoQPupationMax - weight_mg ] ; for queens
```

```
; .. and for worker (or still undefined) larvae:
```

```
if (caste = "worker" or caste = "undefined" or caste = "male") and weight_mg + maxWeightGain_mg
> memoWPupationMax
```

```
[ set maxWeightGain_mg memoWPupationMax - weight_mg ]
```

```
]
```

```
report maxWeightGain_mg
```

```
end
```

```
, *****
;
```

to MergeHedgesProc

; hedges are often represented by a large number of very small patches. If "MinSizeFoodSources?" (and "RemoveEmptyFoodSources?") are switched on, they may only contain one (Average willow) or very few foodsources.

; To avoid this, several small patches of hedges can be merged into a single, larger one (no loss of total area)

; The procedure is called before CreateLayersProc. First, the closest non-hedge food patch for each hedge patch is determined, Then hedges sharing the same closest non-hedge patch are merged by

; summing adding up their areas in one to the biggest of those (~~randomly chosen~~) hedge patches, the other hedge patches get an area of 0 and will be removed in CreateLayersProc.

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

let areaTH_sqm 6500 ; 1000 ; NEW for Beestew! To reduce the risk that hedges are linked to a non-hedge patch which will disappear because it does not provide
; enough resources (area is used as max. nectar and pollen is not defined yet. (The actual removal TH area for Grassland is just below 800m2, other defined SNH types lower.
; for Improved Grassland it is below 6500m2)

let fieldsHedgeLinksList [] ; to link hedges with their closest non-hedge field, format e.g. [[1 17] [5 29] [1 18]..] each sublist with 2 elements: 1st: who of closest non-hedge patch, 2nd: who of hedge

let fieldsWithHedgesList [] ; contains who of all non-hedge patches that are closest to at least one hedge patch

ask

foreach sort-on [(- area_sqm)] foodsources with [patchType = "Hedgerow"] ; NEW for BEESTEW:
; hedges are sorted by area, starting with largest. The largest hedge patch will be kept, all other
; linked hedge patches add their area to it and are removed.

[

let singleHedgeMatchList (list who)

let myField min-one-of foodsources with [patchType != "Hedgerow" and area_sqm >= areaTH_sqm] [distance myself] ; myField saves the (non-hedge) foodsource closest to the current hedge patch

if myField != nobody

[

set singleHedgeMatchList fput [who] of myField singleHedgeMatchList ; this is a 2 item list, 1st item: who of the hedge's closest non-hedge field, second item who of the hedge

set fieldsWithHedgesList lput [who] of myField fieldsWithHedgesList

set fieldsHedgeLinksList lput singleHedgeMatchList fieldsHedgeLinksList

]

]

set fieldsWithHedgesList remove-duplicates fieldsWithHedgesList ; duplicates are removed from the list

foreach fieldsWithHedgesList

[

let myFieldID ?

let hedgesSublist filter [first ? = myFieldID] fieldsHedgeLinksList ; this sublist only contains those elements where the current field is present

let shortSublist []

foreach hedgesSublist [set shortSublist lput (item 1 ?) shortSublist] ; this shortSublist only contains the who of those hedges, linked to the current field

let masterHedgeID -1 ; will save who of the hedge patch that will increase in area

foreach shortSublist

[

ifelse masterHedgeID = -1 ; in this case, the foodsource is the first hedge at that field and will increase in size

[set masterHedgeID ?]

[; the areas of all other hedge patches are now added to the "master" hedge patch

let areaToBeAdded_sqm [area_sqm] of foodsource ?

let nectarToBeAdded_myl [nectarMax_myl] of foodsource ?

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    let pollenToBeAdded_g [pollenMax_g] of foodsource ?
    ask foodsource masterHedgeID
    [
        set area_sqm area_sqm + areaToBeAdded_sqm
        set nectarMax_myl nectarMax_myl + nectarToBeAdded_myl ; if hedges are composed of
layers/several foodsources, this value will be overwritten in CreateLayersProc
        set pollenMax_g pollenMax_g + pollenToBeAdded_g ; if hedges are composed of
layers/several foodsources, this value will be overwritten in CreateLayersProc
    ]
    ask foodsource ?
    [
        set area_sqm 0
        hide-turtle
    ]
]
]
]

ask foodsources with [ patchType = "Hedgerow" ]
[
    set radius_m sqrt (area_sqm / pi) ; [m]
    set size FoodsourceSymbolSizeFactor * radius_m * Scaling_NLpatches/m
    ; if size < MinFoodSourceSymbolSize [ set size MinFoodSourceSymbolSize ]
]

end

, *****

```

to MortalityBroodProc

; Duchateau & Velthuis 1988: "At the end of the colony development the third important phase sets in with the start of worker oviposition and of mutual

; overt aggression between the workers and between the workers and their queen.

In this ultimate phase of the colony development most of the eggs

; and larvae from queen and workers will not survive to adulthood due to the aggressive interactions and to reciprocal oophagy (SLADEN, 1912;

; RICHARDS, 1977; VAN DOORN & HERINGA, 1986; own observations)."

ask bees

```

[
    let memoNumber number ; (either 1 or cohort size)
    if stage = "egg" and broodAge > [ devAgeHatchingMax_d ] of OneSpecies speciesID
    [
        ask colony colonyID [ set eggDeathsIncubation eggDeathsIncubation + memoNumber]
        DieProc "Egg: broodAge > devAgeHatchingMax_d"
    ]
    if stage = "larva"
    [

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    if ((caste = "worker" or caste = "male" or caste = "undefined") and broodAge > [
devAgePupationMax_d] of OneSpecies speciesID )
    or (caste = "queen" and broodAge > [ devAge_Q_PupationMax_d] of OneSpecies speciesID )
    [
        ; Get relative incubation and weights (relative to minimum target required for developing into
the next stage).
        let relativeIncub -1
        let relativeWeight -1
        ifelse caste = "queen"
        [
            set relativeWeight (weight_mg / [ devWeight_Q_PupationMin_mg ] of OneSpecies speciesID)
            set relativeIncub (cumullIncubationReceived_kj / [ devIncubation_Q_PupationTH_kj ] of
OneSpecies speciesID)
        ]
        [
            set relativeWeight (weight_mg / [ devWeightPupationMin_mg ] of OneSpecies speciesID)
            set relativeIncub (cumullIncubationReceived_kj / [ devIncubationPupationTH_kj ] of
OneSpecies speciesID)
        ]
        ; Record the outputs: number of bees that die due to relative weight/incubation received is less
than 1
        if relativeWeight < 1 AND relativeWeight < relativeIncub [ ask colony colonyID [set
larvaDeathsWeight larvaDeathsWeight + memoNumber] ]
        if relativeIncub < 1 AND relativeIncub < relativeWeight [ ask colony colonyID [set
larvaDeathsIncubation larvaDeathsIncubation + memoNumber] ]
        if relativeWeight >= 1 AND relativeIncub >= 1 [ AssertionProc "Neither Weight or Incubation
reason for death: MortalityBroodProc (1)"]
        DieProc "Larva: broodAge > max. pupation age"
    ]
]
if stage = "pupa"
[
    if caste = "undefined" [ AssertionProc "Pupa with undefined caste (MortalityProc)!" ]
    if ((caste = "worker" or caste = "male") and broodAge > [ devAgeEmergingMax_d ] of
OneSpecies speciesID )
    or (caste = "queen" and broodAge > [ devAge_Q_EmergingMax_d ] of OneSpecies speciesID )
    [
        ask colony colonyID ; as pupae are not fed, they died due to lack of incubation
        [ set pupaDeathsIncubation pupaDeathsIncubation + memoNumber ]
        DieProc "Pupa: broodAge > max. emerging age"
    ]
]
if colonyID >= 0 and ticks > [ competitionPointDate ] of Colony colonyID ; development of eggs
into larvae only possible before CP! (Duchateau & Velthuis 1988)
[
    if stage = "egg"
    [
        ask colony colonyID [set broodDeathsCP broodDeathsCP + memoNumber]
        DieProc "Egg: CP!"
    ]
]
]

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
]
end
```

```
, *****
,
```

to NeedNectarPollenLarvaeTodayProc

; calculates how much nectar and pollen is approximately required today to feed the larvae
ask colonies

```
[
  let myColony who
  let pollenNeedMyColony_g 0
  ask bees with [ stage = "larva" and colonyID = myColony ]
    [ set pollenNeedMyColony_g pollenNeedMyColony_g ; pollen need summed up here..
      + number ; calculated from cohort size ..
      * ((MaxWeightGainToday_mg_REP who) ; times max. possible gain in weight..
        / ([pollenToBodymassFactor] of OneSpecies speciesID)) ;..translated into pollen
        / 1000 ] ; units: mg -> g
  set pollenNeedLarvaeToday_g pollenNeedMyColony_g
  set energyNeedToday_kJ pollenNeedLarvaeToday_g *
  EnergyRequiredForPollenAssimilation_kJ_per_g
]
```

end

```
, *****
,
```

to-report NestSiteFoodSourceREP [memoSpecies]

```
let chosenFoodSource nobody
let foodSourceList shuffle sort [nestsiteFoodsourceList] of memoSpecies ; Randomise order of
species-suitable foodSources (actually not necessary!)
let foodSourceArea [nestSiteArea] of memoSpecies ; total area of species-suitable
foodSources
let p random-float 1
let probsSummedUp 0
let foodCounter 0
let fsFound? FALSE
while [not fsFound?]; go through all food sources in the list
[
  let probs [area_sqm / foodSourceArea] of (item foodCounter foodSourceList)
  set probsSummedUp probs + probsSummedUp

  if probsSummedUp > p
  [
    set chosenFoodSource (item foodCounter foodSourceList)
    set fsFound? TRUE
  ]
  set foodCounter foodCounter + 1
]
report chosenFoodSource
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
; *****
```

to NestSitesSearchingProc

```
let memoX 0
let memoY 0
let memoSpecies oneSpecies speciesID
let nestSiteFound false
let memoFoodSource nobody
let dailyChance [chanceFindNest] of memoSpecies ; chance is species-own variable
; Decide if queen finds a nest today
if random-float 1 <= dailyChance
[
; Find FoodSource for colony
set memoFoodSource NestSiteFoodSourceREP memoSpecies
; Find patch for colony
ask memoFoodSource
[
ask one-of patches with [ distance myself < ([radius_m] of myself * SCALING_NLpatches/m) ]
[
set memoX pxcor
set memoY pycor
]
]
set nestSiteFound true
]

ifelse nestSiteFound = true
[
setxy memoX memoY
set activity "nestConstruction"
]
; If nest site not found, queen has probability of dying based on foraging mortality per sec
multiplied by seconds searching for nest site
[
if random-float 1 < 1 - ((1 - MortalityForager_per_s) ^ (NestSearchTime_h * 60 * 60))
; 1 - MortalityForager_per_s: prob. to survive 1s
; ^ (NestSearchTime_h * 60 * 60): prob to survive the searching period
; 1 - prob. to survive = prob. to die

[
DieProc "Queen: died while searching nest site"
]
]
end
```

```
; *****
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

to-report NextActiveBeeREP

; determines which is the next bee to become active

set ActiveBee first ActiveBeesSortedList ; this refers to a bee that in most cases JUST HAD BEEN active! (but not e.g. if the previously active bee just had died!)

let persTime_activeBee [personalTime_s] of bee ActiveBee

; the actual position is somewhere between the minimal and the maximal position:

let minPosition 0 ; counting of items in list start with 0

let maxPosition length ActiveBeesSortedList - 1 ; -1, as counting of items in list start with 0

let currentPosition round (maxPosition / 2) ; don't know where the final position will be so

currentPosition is set to right into the middle

; now the correct position is determined:

while [maxPosition - minPosition > 1]

[

ifelse [personalTime_s] of bee item currentPosition ActiveBeesSortedList > persTime_activeBee

[

set maxPosition currentPosition

set currentPosition round ((currentPosition + minPosition) / 2)

]

[

set minPosition currentPosition

set currentPosition round ((maxPosition + currentPosition) / 2)

]

]

; beginningList is activeList to currentPosition & endList is currentPosition to end of list:

let beginningList sublist ActiveBeesSortedList 0 currentPosition ; beginning to (excluding) currentPosition

let endList sublist ActiveBeesSortedList currentPosition length ActiveBeesSortedList ; from (including) currentPosition to end

; the activeBee is now removed from its original position..

;(if the number of bees in ActiveBeesSortedList is greater than 1, the first bee from the beginningList is removed, as this is the activeBee duplicated

; if number of bees in list is 0, the first bee from the endList is removed, as this is the activeBee duplicated)

ifelse length ActiveBeesSortedList > 1

[set beginningList but-first beginningList] ; first item of beginningList is removed

[set endlist but-first endlist] ; first item of endList is removed

;.. and then the activeBee is placed at its correct position

ifelse (length endList = 1

and persTime_activeBee > [personalTime_s] of bee item 0 endList)

[set ActiveBeesSortedList (sentence beginningList endList ActiveBee)]

[set ActiveBeesSortedList (sentence beginningList ActiveBee endList)]

; the new activeBee is now the first in the list:

report first ActiveBeesSortedList

end

, *****

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

to OutputDailyProc

with-local-randomness ; allows changing/switching off plots without changing the sequence of random numbers

```
[
  random-seed ticks ; local random seed, only valid within this procedure
  set TotalIBMColonies count colonies with [ cohortBased? = false ]
  set TotalQueens sum [ number ] of bees with [ caste = "queen" ]
  set TotalMatedQueens sum [ number ] of bees with [ caste = "queen" and mated? = true ]
  set TotalUnmatedQueens sum [ number ] of bees with [ caste = "queen" and mated? = false ]
  set TotalHibernatingQueens sum [ number ] of bees with [ activity = "hibernate" ]
  set TotalColonies count colonies
  set TotalBeeAgents count bees
  set TotalMales sum [ number ] of bees with [ caste = "male" ]
  set TotalActiveBees length ActiveBeesSortedList
  ifelse count bees with [brood? = false and caste = "worker"] > 0
    [ set MeanWorkerWeight_mg mean [ weight_mg ] of bees with [brood? = false and caste =
"worker"] ]
    [ set MeanWorkerWeight_mg 0 ]
  ifelse count bees with [brood? = false and caste = "queen"] > 0
    [ set MeanQueenWeight_mg mean [ weight_mg ] of bees with [brood? = false and caste =
"queen"] ]
    [ set MeanQueenWeight_mg 0 ]
  ifelse count bees with [brood? = false] > 0
    [ set MeanAdultWeight_mg mean [ weight_mg ] of bees with [brood? = false] ]
    [ set MeanAdultWeight_mg 0 ]
```

```
if TotalMapArea_km2 = 0 and count foodsources > 0 [ set TotalMapArea_km2 TotalMapAreaREP ]
if TotalMapArea_km2 > 0 [ set ColonyDensity_km2 TotalColonies / TotalMapArea_km2 ]
```

```
if ShowPlots? = true
```

```
[
  PlottingProc "plot 1" GenericPlot1 ; PlottingProc is called repeatedly..
```

```
;; REMOVE SEMICOLON TO ADD MORE PLOTS:
```

```
PlottingProc "plot 2" GenericPlot2
PlottingProc "plot 3" GenericPlot3
PlottingProc "plot 4" GenericPlot4
PlottingProc "plot 5" GenericPlot5
]
```

```
if ShowWeather? = true
```

```
[
  ask Signs with [ shape = "sun" ]
  [
    ifelse DailyForagingPeriod_s > 0
      [ show-turtle set label precision (DailyForagingPeriod_s / 3600) 1 ]
      [ hide-turtle set label " " ]
```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
] ; "sun" sign is shown, whenever there is an opportunity to forage
```

```
ask Signs with [ shape = "cloud"]
[
  ifelse DailyForagingPeriod_s < (4 * 3600)
    [ show-turtle ]
    [ hide-turtle ]
] ; "cloud" sign is shown, whenever there is less than 4 hrs of foraging possible
]
]
end
```

```
, *****
```

to PatchesInRangeProc

; called by a colony; creates 2 lists, containing the who of all foodsources and masterpatch-foodsources within the foraging range of the colony

```
let allPatches []
let allMasterPatches []
let xcol xcor
let ycol ycor
set allPatchesInRangeList []
set masterpatchesInRangeList []
ask foodsources with [ distancexy xcol ycol <= (ForagingRangeMax_m * Scaling_NLpatches/m ) ]
[
  set allPatches fput who allPatches
  if masterpatch? = true [ set allMasterPatches fput who allMasterPatches ]
]
set allPatchesInRangeList allPatches ; rather: all food sources in range
set masterpatchesInRangeList allMasterPatches
end
```

```
, *****
```

to PlottingProc [plotname plotChoice]

```
set-current-plot plotname
if plotChoice = "Foodsources sizes (histogram)"
[
  set-plot-x-range 0 10
  create-temporary-plot-pen "N "
  set-plot-pen-mode 1 ; 1: bars
  set-plot-pen-color black
  set-plot-pen-interval 1
  histogram [ size ] of Foodsources
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

if plotChoice = "Matrilines (histogram)" ; NOTE: this plot does NOT correct for "number" (cohort size), hence IBM colonies will be overrepresented!

```
[
  set-plot-x-range 0 140
  create-temporary-plot-pen "mtGene"
  set-plot-pen-mode 1 ; 1: bars
  set-plot-pen-color black
  set-plot-pen-interval 0.1
  histogram [ mtDNA ] of bees with [ caste = "queen" ]
]
```

if plotChoice = "Genepool (histogram)" ; NOTE: this plot does NOT correct for "number" (cohort size), hence IBM colonies will be overrepresented!

```
[
  let genepool []
  ask bees with [ caste = "queen" ]
  [
    foreach allelesList
      [ set genepool fput ? genepool ]
    foreach spermathecalList
      [ set genepool fput ? genepool ]
  ]
  set-plot-x-range 0 140
  create-temporary-plot-pen "alleles"
  set-plot-pen-mode 1 ; 1: bars
  set-plot-pen-color black
  set-plot-pen-interval 0.1
  histogram genepool ;
]
```

if plotChoice = "Colony sizes (histogram)" and count Colonies > 0 ; NOTE: this plot does NOT correct for "number" (cohort size), hence IBM colonies will be overrepresented!

```
[
  if (max [colonysize] of Colonies > 0)
  [
    set-plot-x-range 0 10
    set-plot-x-range 0 max [colonysize] of Colonies
    create-temporary-plot-pen "N "
    set-plot-pen-mode 1 ; 1: bars
    set-plot-pen-color black
    set-plot-pen-interval 20
    histogram [ colonysize ] of Colonies
  ]
]
```

if plotChoice = "Bee weights [mg] (histogram)" ; NOTE: this plot does NOT correct for "number" (cohort size), hence IBM colonies will be overrepresented!

```
[
  create-temporary-plot-pen "queens"
  set-plot-pen-color red
  set-plot-x-range 0 1500
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

set-plot-y-range 0 40
set-plot-pen-mode 1 ; 1: bars
set-plot-pen-interval 50
histogram [ weight_mg ] of bees with [brood? = false and caste = "queen"]

create-temporary-plot-pen "workers"
set-plot-pen-color black
set-plot-pen-mode 1 ; 1: bars
set-plot-pen-interval 50
histogram [ weight_mg ] of bees with [brood? = false and caste = "worker"]

create-temporary-plot-pen "males"
set-plot-pen-color green
set-plot-pen-mode 1 ; 1: bars
set-plot-pen-interval 50
histogram [ weight_mg ] of bees with [brood? = false and caste = "male"]
]

```

```

if plotChoice = "NNumber of colonies"
[
  set-plot-x-range 0 10
  create-temporary-plot-pen "Cols"# colonies"
  plotxy ticks count Colonies
]

```

```

if plotChoice = "Species N coloniesNumber of adult queens"
-[
- set-plot-x-range 0 10

- create-temporary-plot-pen "B\_terrestris"# queens"
plotxy ticks TotalAdultQueens
_]

```

```

if plotChoice = "Number of colonies for different species"
__[
__ let col 5
__ set-plot-x-range 0 10
__ foreach SpeciesList
__ [
__ create-temporary-plot-pen ?
__ set-plot-pen-color yellowcol
__ plotxy ticks count colonies with [shape = "b\_terrestris"]speciesNameColony = ? ]

__ create-temporary-plot-pen "B_lapidarius"
__ set-plot-pen-color black
__ plotxy ticks count colonies with [shape = "b_lapidarius"]

__ create-temporary-plot-pen "B_pascuorum"
__ set-plot-pen-color brown col col + 10

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
—plotxy ticks count colonies with [shape = "b_pascuorum"]
```

```
—create-temporary-plot-pen "B_hortorum"
```

```
—set-plot-pen-color green
```

```
—plotxy ticks count colonies with [shape = "b_hortorum"]
```

```
—create-temporary-plot-pen "B_pratorum"
```

```
- _]
```

```
_]
```

```
==set-plot-pen-color orange
```

```
==plotxy ticks count colonies with [shape = "b_pratorum"]
```

```
—create-temporary-plot-pen "B_hypnorum"
```

```
—set-plot-pen-color blue
```

```
—plotxy ticks count colonies with [shape = "b_hypnorum"]
```

```
—}
```

```
if plotChoice = "Foraging period max. [hrs]"
```

```
[
```

```
  set-plot-x-range 0 10
```

```
  create-temporary-plot-pen "max. foraging"
```

```
  plotxy ticks DailyForagingPeriod_s / 3600
```

```
]
```

```
if plotChoice = "Foraging trips daily"
```

```
[
```

```
; set-plot-x-range 0 10
```

```
  create-temporary-plot-pen "N trips total"
```

```
  plotxy ticks TotalForagingTripsToday
```

```
]
```

```
if plotChoice = "Food available"
```

```
[
```

```
; set-plot-x-range 0 10
```

```
  create-temporary-plot-pen "Nectar_l"
```

```
  set-plot-pen-color yellow
```

```
  plotxy ticks NectarAvailableTotalNectarAvailableTotalToday_l
```

```
  create-temporary-plot-pen "Pollen_kg"
```

```
  set-plot-pen-color red
```

```
  plotxy ticks PollenAvailableTotalPollenAvailableTotalToday_kg
```

```
]
```

```
if plotChoice = "Total adults"
```

```
[
```

```
  set-plot-x-range 0 10
```

```
  create-temporary-plot-pen "Adults"
```

```
  plotxy ticks TotalAdults
```

```
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
if plotChoice = "Species total adults"
[
  let col 5
  set-plot-x-range 0 10
  foreach SpeciesList
  [
    create-temporary-plot-pen "B_terrestris"?
    set-plot-pen-color yellowcol
    plotxy ticks sum [ number ] of bees with [ speciesName = "B_terrestris"? and brood? = false ]::
    and colonyID > 0 ]

  set col col + 10
  ]
]

if plotChoice = "Species total adult queens"
[
  let col 5
  set-plot-x-range 0 10
  foreach SpeciesList
  [
    create-temporary-plot-pen "B_lapidarius"?
    set-plot-pen-color blackcol
    plotxy ticks sum [ number ] of bees with [ speciesName = "B_lapidarius"? and brood? = false
    and colonyID > 0 caste = "queen" ]

    set col col + 10
  ]
]

if plotChoice = "Species hibernating queens"
[
  let col 5
  set-plot-x-range 0 10
  foreach SpeciesList
  [
    create-temporary-plot-pen "B_pascuorum"?
    set-plot-pen-color browncol
    plotxy ticks sum [ number ] of bees with [ speciesName = "B_pascuorum"? and brood? = false
    and colonyID > 0 ]

    create-temporary-plot-pen "B_hortorum"
    set-plot-pen-color green
    plotxy ticks sum [ number ] of bees with [ speciesName = "B_hortorum" and brood? = false and
    colonyID > 0 ]

    create-temporary-plot-pen "B_pratorum"
    set-plot-pen-color orange
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_pratorum" and brood? = false and colonyID > 0 ]
```

```
—create temporary plot pen "B_hypnorum"
```

```
—set plot pen color blue
```

```
—plotxy ticks count bees with [ speciesName = "B_hypnorum" and brood? = false and colonyID > 0 ]
```

```
—create temporary plot pen "Psithyrus"
```

```
—set plot pen color red
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "Psithyrus" and brood? = false and colonyID > 0 ]
```

```
—}
```

```
—if plotChoice = "Species total adult queens"
```

```
—{
```

```
—set plot x range 0 10
```

```
—create temporary plot pen "B_terrestris"
```

```
—set plot pen color yellow
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_terrestris" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "B_lapidarius"
```

```
—set plot pen color black
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_lapidarius" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "B_pascuorum"
```

```
—set plot pen color brown
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_pascuorum" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "B_hortorum"
```

```
—set plot pen color green
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_hortorum" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "B_pratorum"
```

```
—set plot pen color orange
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_pratorum" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "B_hypnorum"
```

```
—set plot pen color blue
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "B_hypnorum" and brood? = false and caste = "queen" ]
```

```
—create temporary plot pen "Psithyrus"
```

```
—set plot pen color red
```

```
—plotxy ticks sum [ number ] of bees with [ speciesName = "Psithyrus" and brood? = false and caste = "queen" ]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

—}

~~—if plotChoice = "Species hibernating queens"~~

~~—{~~

~~—set-plot-x-range 0 10~~

~~—create-temporary-plot-pen "B_terrestris"~~

~~—set-plot-pen-color yellow~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_terrestris" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "B_lapidarius"~~

~~—set-plot-pen-color black~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_lapidarius" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "B_pascuorum"~~

~~—set-plot-pen-color brown~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_pascuorum" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "B_hortorum"~~

~~—set-plot-pen-color green~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_hortorum" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "B_pratorum"~~

~~—set-plot-pen-color orange~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_pratorum" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "B_hypnorum"~~

~~—set-plot-pen-color blue~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "B_hypnorum" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—create-temporary-plot-pen "Psithyrus"~~

~~—set-plot-pen-color red~~

~~—plotxy-ticks-sum [number] of bees with [speciesName = "Psithyrus" and brood? = false and caste = "queen" and activity = "hibernate"]~~

~~—set col col + 10~~

~~—]~~

~~_]~~

if plotChoice = "Hibernating queens"

[

set-plot-x-range 0 10

create-temporary-plot-pen "N "

plotxy ticks sum [number] of bees with [activity = "hibernate"]

]

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWART: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
if plotChoice = "Egg-layingTotal eggs"
[
  set-plot-x-range 0 10
  set-plot-x-range 0 10
  create-temporary-plot-pen "periodN"
  ifelse ShowInspectedColony? = true
  [
    ifelse count colonies with [ who = InspectTurtle ] = 1
    [ plotxy ticks [allEggs] of Colony InspectTurtle ]
    [ clear-plot ]
  ]
  [ plotxy ticks (TotalEggs) ]
]

if plotChoice = "Stores: honeynectar [ml] & pollen [g]" ; and count Colonies > 0
[
  set-plot-x-range 0 10
  set-plot-x-range 0 10
  create-temporary-plot-pen "honeynectar"
  set-plot-pen-color yellow

  ifelse count Colonies = 0
  [ plotxy ticks 0 ]
  [
    ifelse ShowInspectedColony? = true
    [
      ifelse count colonies with [ who = InspectTurtle ] = 1
      [ plotxy ticks [energyStore_kJ] of Colony InspectTurtle / EnergyHoney_kJ/ml ]
      [ clear-plot ]
    ]
    [ plotxy ticks (mean [ energyStore_kJ ] of Colonies) / EnergyHoney_kJ/ml ]
  ]

  create-temporary-plot-pen "pollen"
  set-plot-pen-color orange
  ifelse count Colonies = 0
  [ plotxy ticks 0 ]
  [
    ifelse ShowInspectedColony? = true
    [
      ifelse count colonies with [ who = InspectTurtle ] = 1
      [ plotxy ticks [pollenStore_g] of Colony InspectTurtle ]
      [ clear-plot ]
    ]
    [ plotxy ticks (mean [ pollenStore_g ] of Colonies) ]
  ]
]
```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
if plotChoice = "Colony structures"
[
  ifelse ShowInspectedColony? = true
  [
    ifelse count colonies with [ who = InspectTurtle ] = 1
    [
      set-plot-x-range 0 10
      create-temporary-plot-pen "Eggs"
      set-plot-pen-color blue
      plotxy ticks [allEggs] of Colony InspectTurtle
      create-temporary-plot-pen "Larvae"
      set-plot-pen-color yellow
      plotxy ticks [allLarvae] of Colony InspectTurtle
      create-temporary-plot-pen "Pupae"
      set-plot-pen-color brown
      plotxy ticks [allPupae] of Colony InspectTurtle
      create-temporary-plot-pen "Workers"
      set-plot-pen-color black
      plotxy ticks [allAdultWorkers] of Colony InspectTurtle
      create-temporary-plot-pen "Males"
      set-plot-pen-color green
      plotxy ticks [allAdultMales] of Colony InspectTurtle
      create-temporary-plot-pen "Queens"
      set-plot-pen-color red
      plotxy ticks [allAdultQueens] of Colony InspectTurtle
    ]
    [ clear-plot ] ; plot is cleared after the previous 'inspected colony' has died
  ]
]

[ ; if ShowInspectedColony? = FALSE:
  set-plot-x-range 0 10
  create-temporary-plot-pen "Eggs"
  set-plot-pen-color blue
  plotxy ticks TotalEggs
  create-temporary-plot-pen "Larvae"
  set-plot-pen-color yellow
  plotxy ticks TotalLarvae
  create-temporary-plot-pen "Pupae"
  set-plot-pen-color brown
  plotxy ticks TotalPupae
  create-temporary-plot-pen "Workers"
  set-plot-pen-color orange
  plotxy ticks TotalAdultWorkers
  create-temporary-plot-pen "Males"
  set-plot-pen-color green
  plotxy ticks TotalAdultMales
  create-temporary-plot-pen "Queens"
  set-plot-pen-color black
  plotxy ticks TotalAdultQueens
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

]

if plotChoice = "Switchpoints"

[

set-plot-x-range 0 50

create-temporary-plot-pen "SP"

set-plot-pen-color black

set-plot-pen-mode 1

histogram [switchPointDate - eusocialPhaseDate] of Colonies with [eusocialPhaseDate + switchPointDate < NotSetHigh]

]

if plotChoice = "Sex ratio"

[

set-plot-y-range 0 12

create-temporary-plot-pen "M:FQ"

set-plot-pen-color black

set-plot-pen-mode 0

if TotalAdultQueens > 0

[

plot TotalAdultMales / TotalAdultQueens

]

]

if plotChoice = "Tongue lengths workers (histogram)"

[

set-plot-x-range 0 20

create-temporary-plot-pen "tongue"

set-plot-pen-color black

set-plot-pen-mode 1

histogram [glossaLength mm] of bees with [caste = "worker" and stage = "adult"]

]

if plotChoice = "Tongue lengths queens (histogram)"

[

set-plot-x-range 0 20

create-temporary-plot-pen "tongue"

set-plot-pen-color black

set-plot-pen-mode 1

histogram [glossaLength mm] of bees with [caste = "queen" and stage = "adult"]

]

if plotChoice = "Biomass (dw) [kg]"

[

set-plot-x-range 0 10

;set-plot-y-range 0 0.1

create-temporary-plot-pen "biomass"

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

plotxy ticks sum [weight mg] of bees * mean [number] of bees / 1000000 / 2.64 ; 2.64: derived from Sage 1982, Tab. 1: mean wet:dry weight for diptera/hymenoptera for lengths of 2 - 22mm (see SB_Submodels.xlsx

_]

if plotChoice = "Biomass (dw) [kg] / AbundanceBoost" and TotalBeeAgents > 0

_]

set-plot-x-range 0 10

;set-plot-y-range 0 1

create-temporary-plot-pen "biomass"

plotxy ticks (sum [weight mg] of bees * mean [number] of bees / 1000000 / 2.64) /

AbundanceBoost ; 2.64: derived from Sage 1982, Tab. 1: mean wet:dry weight for

diptera/hymenoptera for lengths of 2 - 22mm (see SB_Submodels.xlsx

_]

if plotChoice = "Age distributions bee agents (histogram)" ; NOTE: this plot does NOT correct for "number" (cohort size), hence IBM colonies will be overrepresented!

_]

create-temporary-plot-pen "adults"

set-plot-pen-color black

set-plot-x-range 0 500

set-plot-y-range 0 40

set-plot-pen-mode 1 ; 1: bars

set-plot-pen-interval 20

histogram [adultAge] of bees with [brood? = false]

_]

create-temporary-plot-pen "brood"

set-plot-pen-color orange

set-plot-pen-mode 1 ; 1: bars

set-plot-pen-interval 20

histogram [broodAge] of bees with [brood? = true]

_]

end

, *****

,

to-report ProboscisLengthREP

let minWeight_mg [devWeightPupationMin_mg] of oneSpecies speciesID

let maxWeight_mg [devWeight_Q_PupationMax_mg] of oneSpecies speciesID

let minLength_mm [proboscis_min_mm] of oneSpecies speciesID

let maxLength_mm [proboscis_max_mm] of oneSpecies speciesID

let slope (maxLength_mm - minLength_mm) / (maxWeight_mg - minWeight_mg)

let proboscisLength_mm minLength_mm + (weight_mg - minWeight_mg) * slope

if weight_mg < minWeight_mg or weight_mg > maxWeight_mg [AssertionProc ("Wrong bee weight in ProboscisLengthREP Min")]

report proboscisLength_mm

end

, *****

,

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

to PsithyrusNestSearchProc

```
; determines if a cuckoo bee finds a suitable host colony and can enter it successfully
let memoColID -1
let findSingleNestProb 0.05 ; arbitrary value
let getAccessProb 0.25 ; arbitrary value
let getKilledProb 0.25 ; arbitrary value
let killQueenProb 0.5 ; arbitrary value
let succesful false
let myWho who
let findAnyNestProb 1 - ((1 - findSingleNestProb) ^ count colonies)

if random-float 1 < findAnyNestProb
[
  if random-float 1 < getAccessProb
  [
    ifelse random-float 1 < getKilledProb
    [ DieProc "Psithyrus: killed by Bombus queen" ]
    [
      set succesful true
      set color black
      set size size * 8
      set shape "circleSingle"
      ask one-of colonies [ set memoColID who ]
      set colonyID memoColID
      move-to colony colonyID
      set thForagingNectar ThresholdLevelREP "nectarForaging" "Psith"
      set thForagingPollen ThresholdLevelREP "pollenForaging" "Psith"
      set thNursing ThresholdLevelREP "nursing" "Psith"
      set thEggLaying ThresholdLevelREP "eggLaying" "Psith"
    ]
  ]
]
ifelse succesful = true
[
  if count bees with [colonyID = memoColID and caste = "queen" and mated? = true] > 0
  [
    ask bees with [colonyID = memoColID and caste = "queen" and mated? = true and who !=
myWho ] ; queen might be killed by Psithyrus
    [
      if random-float 1 < killQueenProb
      [
        DieProc "Queen killed by cuckoo bee!"
      ]
    ]
  ]
]
; If not successful at finding a nest
[
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    if random-float 1 < 1 - ((1 - MortalityForager_per_s) ^ (NestSearchTime_h * 60 * 60))
    [
        DieProc "Psithyrus: died while searching nest"
    ]
]
end

```

, *****

to QueenProductionDateProc

```

; for B. terrestris, based on Duchateau & Velthuis 1988 - no data for other species!
; queenProductionDate: the (theoretical) date (time step) when the first queen-destined eggs were
laid, back calculated on the day when "queenProduction?" is set true (in QueenProductionDateProc).
; Does not require that any eggs were actually laid on that day.
ask colonies
[
    let memoColony who
    let averageCumulTimeEgg_d 7 ; average (realised) duration of egg phase in the model is about 6.4;
set to 7 as this results in better sex ratio than 6 (min hatching age 5, max hatching age 10)
    let timeEggToLarvalAgeAtQueenDetermination averageCumulTimeEgg_d + [
dev_larvalAge_QueenDetermination_d ] of OneSpecies speciesIDcolony ; i.e. 7+3=10d for B.
terrestris
    ; (only) if larvae of the right age are present, timeEggToLarvalAgeAtQueenDetermination can be
directly determined from their brood age (in this case, the previous value is overwritten)
    if any? bees with [ stage = "larva" and colonyID = memoColony and cumulTimeLarva_d = [
dev_larvalAge_QueenDetermination_d ] of OneSpecies speciesID ]
    [ set timeEggToLarvalAgeAtQueenDetermination max[ broodAge ] of bees with [ stage = "larva"
and colonyID = memoColony and cumulTimeLarva_d = [ dev_larvalAge_QueenDetermination_d ] of
OneSpecies speciesID ] ]
    ; asking for "max" in case there are 2 larval cohorts of dev_larvalAge_QueenDetermination_d
age but different broodAges (because younger cohort has developed quicker as eggs).
    ; This should not happen in the current version, but might be the case in a future version.
    if queenProduction? = false ; is set true ca. 5 - 16 days after SW was determined
    and ticks - timeEggToLarvalAgeAtQueenDetermination ; this is the date when the larvae which
are today at the queen determination stage were laid as eggs
    >= switchPointDate - QueenDestinedEggsBeforeSP_d ; "queen eggs" are laid
QueenDestinedEggsBeforeSP_d (5d) before switchpoint at earliest
    and larvaWorkerRatio < LarvaWorkerRatioTH ; but also the L:W ratio on that day has to be
below LarvaWorkerRatioTH (= 3)
    [
        set queenProduction? true ; female larvae can now develop into queens
        set queenProductionDate ticks - timeEggToLarvalAgeAtQueenDetermination ; ..these larvae
were laid as eggs on that day
    ]
]
end

```

, *****

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

to QueensLeavingNestProc

```
; young queens leave the colony, mate and hibernate:
ask bees with [ stage = "adult" and caste = "queen" and mated? = false and colonyID != -1 ] ; young,
adult queens still in a colony..
[
  let memoSpecies speciesID
  ifelse count bees with [ caste = "male" and stage = "adult" and speciesID = memoSpecies ] > 0 ; if
suitable males are present..
  [
    let newAlleleList [] ; .. the queen will mate with one
    ask one-of bees with [ caste = "male" and stage = "adult" and speciesID = memoSpecies ] ;
mating with a haploid or diploid(!) adult male of the same species,
    [ set newAlleleList allelesList ]
    set spermathecaList newAlleleList
  ]
  [ ; mating with a male from outside:
    if UnlimitedMales? = false [ DieProc "Queen: no mating" ] ; if queen's can't mate, they are
removed
    let foreignAllele -1 * (random N_ForeignAlleles) - 1 ; random integer number: -1, -2, ... -
N_ForeignAlleles
    set spermathecaList fput foreignAllele spermathecaList
    if length spermathecaList > 1 [ AssertionProc "Assertion violated in QueensLeavingNestProc: too
many alleles here!" ]
  ]
  set mated? true ; queen is now mated
  set thEgglaying ThresholdLevelREP "eggLaying" "QueenInitiationPhase" ; queen is now ready to
lay eggs
  set size QueenSymbolSize
  set shape "circle"
  set color red
  set activity "hibernate" ; queen hibernates and be active untile she emerges in spring
  set colonyID -1 ; queen is no longer member of a colony
  if length spermathecaList = 2 [ DieProc "Queen: mating with diploid male" ]
  ; queens mating with diploid male are removed from the simulation as they are not able to
establish a colony (Duchateau & Marien 1995)
]
end
```

, *****

to-report StimEgglayingREP

```
; calculates the stimulus for egg laying within a colony
let egglayingStim 0
if ((pollenStore_g > [ minPollenStore_g ] of oneSpecies speciesIDcolony
and (allEggs + allLarvae) = 0))
or allAdults > 1 ; i.e. if at least 1 worker is present
[ set egglayingStim 1 ]
report egglayingStim
end
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
, *****
```

to-report StimForagingNectarREP

```
; asked by colony
let storeSize_d 5
let minNectarStore_kJ 20
let nectarStimTH 0.005 ; heuristically determined
let idealEnergyFactor 6 ; heuristically determined
set idealEnergyStore_kJ idealEnergyFactor * energyNeedToday_kJ * storeSize_d +
minNectarStore_kJ
if idealEnergyStore_kJ < 0 [ AssertionProc "Negative idealEnergyStore_kJ!
(StimForagingNectarREP)" ]
let nectarStim (idealEnergyStore_kJ - energyStore_kJ) / idealEnergyStore_kJ
ifelse nectarStim > nectarStimTH
[ set nectarStim 1 ]
[ set nectarStim 0 ]
if (Daytime_s < Sunrise_s) or (Daytime_s > Sunrise_s + DailyForagingPeriod_s) ; foraging only
during daytime
[ set nectarStim 0 ]
report nectarStim
end
```

```
, *****
```

to-report StimForagingPollenREP

```
; asked by colony
let storeSize_d 5
let pollenStimTH 0.005 ; heuristically determined
set idealPollenStore_g pollenNeedLarvaeToday_g * storeSize_d + [ minPollenStore_g ] of
oneSpecies speciesIDcolony
if idealPollenStore_g < 0 [ AssertionProc "Negative idealPollenStore_kJ! (StimForagingPollenREP)" ]
let pollenStim (idealPollenStore_g - pollenStore_g) / idealPollenStore_g
ifelse pollenStim > pollenStimTH
[ set pollenStim 1 ]
[ set pollenStim 0 ]
if (Daytime_s < Sunrise_s) or (Daytime_s > Sunrise_s + DailyForagingPeriod_s) ; foraging only
during daytime
[ set pollenStim 0 ]
report pollenStim
end
```

```
, *****
```

to-report StimNursingREP

```
; asked by colony
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
; if the energy required for incubation today (devQuotaIncubationToday_kJ), which is the average
energy required per day
; for an individual (devQuotaIncubationToday_kJ) times the individual brood (allEggs + allLarvae +
allPupae) is smaller then
; the incubation actually received today (), then the stimulus to nurse (nursingStim) is set to 1 or
otherwise to 0.
```

```
let nursingStim 0
let incubationRequiredToday_kJ [ devQuotaIncubationToday_kJ ] of Onespecies speciesIDcolony *
(allEggs + allLarvae + allPupae) ; approx. incubation required for whole brood nest today
set nursingStim 0
if incubationRequiredToday_kJ > summedIncubationToday_kJ [ set nursingStim 1 ] ; bees will try to
incubate brood nest, until requirements for today are fulfilled
report nursingStim
```

```
end
```

```
, *****
```

to-report ThresholdLevelREP [thType situation]

```
;TYPES: eggLaying pollenForaging nectarForaging nursing
```

```
;SITUATIONS: egg worker youngQueen QueenInitiationPhase QueenSocialPhase Psith
```

```
let th -1
;Egg Laying
if thType = "eggLaying"
[
  if situation = "egg" [set th NotSetHigh]
  if situation = "worker" [set th NotSetHigh]
  if situation = "youngQueen" [set th NotSetHigh]
  if situation = "QueenInitiationPhase" [set th 0.1]
  if situation = "QueenSocialPhase" [set th 0]
  if situation = "Psith" [set th 0.2]
]
```

```
; Pollen foraging
if thType = "pollenForaging"
[
  if situation = "egg" [set th NotSetHigh]
  if situation = "worker" [set th 0.9]
  if situation = "youngQueen" [set th NotSetHigh]
  if situation = "QueenInitiationPhase" [set th 0.7]
  if situation = "QueenSocialPhase" [set th NotSetHigh]
  if situation = "Psith" [set th NotSetHigh]
]
```

```
; Nectar foraging
if thType = "nectarForaging"
[
  if situation = "egg" [set th NotSetHigh]
```


Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
    if situation = "worker"          [set th 0.9]
    if situation = "youngQueen"      [set th NotSetHigh]
    if situation = "QueenInitiationPhase" [set th 0.7]
    if situation = "QueenSocialPhase" [set th NotSetHigh]
    if situation = "Psith"           [set th NotSetHigh]
  ]

; Nursing
if thType = "nursing"
[
  if situation = "egg"          [set th NotSetHigh]
  if situation = "worker"       [set th 0.9]
  if situation = "youngQueen"   [set th NotSetHigh]
  if situation = "QueenInitiationPhase" [set th 0.5]
  if situation = "QueenSocialPhase" [set th 0.9]
  if situation = "Psith"        [set th NotSetHigh]
]
if th = -1 [ AssertionProc "Assertion violated in ThresholdLevelREP: TH not set!" ]
report th
end

, *****
;
```

to-report TotalMapAreaREP

```
let topSource max-one-of foodsources [ycor]
let bottomSource min-one-of foodsources [ycor]
let rightSource max-one-of foodsources [xcor]
let leftSource min-one-of foodsources [xcor]

let topCor [ycor] of topSource + ([radius_m] of topSource * Scaling_NLpatches/m)
let bottomCor [ycor] of bottomSource - ([radius_m] of bottomSource * Scaling_NLpatches/m)
let rightCor [xcor] of rightSource + ([radius_m] of rightSource * Scaling_NLpatches/m)
let leftCor [xcor] of leftSource - ([radius_m] of leftSource * Scaling_NLpatches/m)

if topCor > max-pycor [ set topCor max-pycor ]
if bottomCor < min-pycor [ set bottomCor min-pycor ]
if rightCor > max-pxcor [ set rightCor max-pxcor ]
if leftCor < min-pxcor [ set leftCor min-pxcor ]

let area_m2 ((topCor - bottomCor) * (rightCor - leftCor)) / (Scaling_NLpatches/m ^ 2)
report area_m2 / (1000 * 1000) ; report real area of map in km2
end
```

```
, *****
;
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

to UpdateColoniesProc

```
ask colonies
[
  let whoCol who
  let countBroodMort_NA 0 ;count number of brood dying through no adults left
  let countBroodMort_ES 0 ;count number of brood dying through energy stores being empty
  if FoodsourcesInFlowerUpdate? = true
    [ FoodsourcesInFlowerAndRangeProc ] ; updated, if some foodsources started or stopped
flowering today
  if ticks > competitionPointDate ; death of colony after competition point
    and allEggs + allLarvae + allPupae = 0
    [
      ask bees with [ colonyID = whoCol and adultAge > 10 and (caste = "worker" or caste =
"queen")]; as males are outside the colony they are killed separately in UpdateSeasonalEventsProc
      [ DieProc "Colony death after CP!" ]
    ]
  if energyStore_kJ <= 0
  [
    ask bees with [ colonyID = whoCol ]
    [
      if brood? = TRUE [set countBroodMort_ES countBroodMort_ES + number]
      DieProc "Colony's energy store depleted!"
    ]
  ]
  if (sum [ number ] of bees with [ colonyID = whoCol and brood? = false ] = 0)
  [ ask bees with [ colonyID = whoCol ]
    [
      if brood? = TRUE [set countBroodMort_NA countBroodMort_NA + number]
      DieProc "No adult bees left!"
    ]
  ]
  set broodDeathsNoAdults broodDeathsNoAdults + countBroodMort_NA
  set broodDeathsEnergyStores broodDeathsEnergyStores + countBroodMort_ES
  set summedIncubationToday_kJ 0

; STATS:
  set allEggs sum [ number ] of bees with [ colonyID = whoCol and stage = "egg" ]
  set allLarvae sum [ number ] of bees with [ colonyID = whoCol and stage = "larva" ]
  set allPupae sum [ number ] of bees with [ colonyID = whoCol and stage = "pupa" ]
  set allAdults sum [ number ] of bees with [ colonyID = whoCol and stage = "adult" ]
  set allAdultWorkers sum [ number ] of bees with [ colonyID = whoCol and caste = "worker" and
stage = "adult" ]
  set allAdultQueens sum [ number ] of bees with [ colonyID = whoCol and caste = "queen" and
brood? = false ]
  set allAdultActiveQueens sum [ number ] of bees with [ colonyID = whoCol and caste = "queen"
and activity != "hibernate" and brood? = false ]
  set allAdultMales sum [ number ] of bees with [ colonyID = whoCol and caste = "male" and brood?
= false ]
  set colonySize sum [ number ] of bees with [ colonyID = whoCol ]
  ifelse allAdultWorkers > 0
    [ set larvaWorkerRatio allLarvae / allAdultWorkers ]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
[ set larvaWorkerRatio NotSetHigh ]
set colonyWeight_mg sum [number * weight_mg] of bees with [ colonyID = whoCol ]

; determine eusocial phase (starts with emergence of first worker):
if eusocialPhaseDate = NotSetHigh and allAdultWorkers > 0
  [ set eusocialPhaseDate ticks ]

; determine SWITCH POINT:
if switchPointDate = NotSetHigh ; i.e. the colony/queen hasn't switched to lay haploid eggs
[
  if eusocialPhaseDate < NotSetHigh ; i.e. colony is now in the eusocial phase
  and larvaWorkerRatio < LarvaWorkerRatioTH
  [
    if random-float 1 <= DailySwitchProbability ; DailySwitchProbability affects the sex ratio:
    increasing it results in more males, decreasing it in more queens
    [ set switchPointDate ticks ]
  ]
]

; determine COMPETITION POINT and TERMINATE colony:
if competitionPointDate = NotSetHigh
  and eusocialPhaseDate < NotSetHigh
  and queenProductionDate < NotSetHigh
[ set competitionPointDate CompetitionPointDateREP ]
if ticks >= competitionPointDate [ set heading 180 ] ; colony symbol is turned on its head after CP

; LABELS & SIGNS:
set label colonysize
if count bees with [ colonyID = whoCol and caste = "queen" and mated? = true ] = 0
  [ set queenright? false ]

ifelse count bees with [ colonyID = whoCol ] = 0
[
  ; to display the production of reproductives on the map..
  let malesHere totalMalesProduced ; ... the numer of adult males..
  let queensHere totalQueensProduced ; .. and adult queens ever produced by this dying colony..
  ask patch-here
  [
    set nMalesProduced nMalesProduced + malesHere ; .. is added to the total number of
    males..
    set nQueensProduced nQueensProduced + queensHere ; and queens ever produced here at
    this Netlogo patch
    set nColonies nColonies + 1 ; ..and the total colonies here
  ]
  let endSeasonDate [seasonStop] of onespecies speciesIDcolony
  ifelse day >= EndSeasonDate
  [ set ColonyDeathsEndSeason ColonyDeathsEndSeason + 1 ]
  [ set ColonyDeathsNoBees ColonyDeathsNoBees + 1 ]
  DieProc "Colony: No adults or brood left in this colony!" ; colony dies, as no bees are left
]
[
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```

    set colonyAge colonyAge + 1
  ]
]
if KeepDeadColonies? = false and Day = 1 [ ask DeadCols [ die ] ] ; dead colonies can be removed
from the simulation with the new year
end

```

, *****

to UpdateColonyStoreBarsProc

```

ask storeBars
[
  let nectarSizeFactor 0
  let pollenSizeFactor 0
  ifelse colony storeColonyID = nobody
  [ die ] ; storeBars die here (and not in DieProc as not a biological agent)
  [
    ask colony storeColonyID
    [
      set nectarSizeFactor energyStore_kJ / (idealEnergyStore_kJ + 0.00001) ; + 0.00001 to avoid
division by zero
      set pollenSizeFactor pollenStore_g / (idealPollenStore_g + 0.00001)
      if nectarSizeFactor > 1 [ set nectarSizeFactor 1 ]
      if pollenSizeFactor > 1 [ set pollenSizeFactor 1 ]
    ]
  ]
  if store = "Nectar"
  [ set size maxSize * nectarSizeFactor ]
  if store = "Pollen"
  [ set size maxSize * pollenSizeFactor ]
]
end

```

, *****

to UpdateFoodsourcesProc

```

; updating FOODSOURCES (nectar & pollen):
set PollenAvailableTotalPollenAvailableTotalToday_kg 0
set NectarAvailableTotalNectarAvailableTotalToday_l 0
set FoodsourcesInFlowerUpdate? false ; might be set true below (if true, colonies need to update
their nectarInFlowerAndRangeList)
ask foodsources
[
  if startDay > StopDay ; turn of the year must not occur during the flowering period of any food
source!
  [ AssertionProc "Foodsource: startDay > StopDay! (UpdateFoodsourcesProc)" ]
  if day = startDay or day = stopDay [ set FoodsourcesInFlowerUpdate? true ] ; if true, colonies need
to update their nectarInFlowerAndRangeList

```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

ifelse day >= startDay and day < StopDay ; IMPRORTANT: day < StopDay (and NOT <=) otherwise nectarInFlowerAndRangeList is not updated correctly!!

```
[
  set nectar_myl nectarMax_myl
  set pollen_g pollenMax_g
]
[
  set nectar_myl 0
  set pollen_g 0
]
set NectarAvailableTotalToday | NectarAvailableTotalToday | + (nectar_myl / (1000 * 1000))
set PollenAvailableTotalToday_kg PollenAvailableTotalToday_kg + (pollen_g / 1000)
]
end
```

; *****

to UpdateMorning_Proc

```
if AssertionViolated [ ask patches [ set pcolor red ] user-message "Assertion violated!" ]
set Day round (ticks mod 365.00000001)
if ticks > 0 and (Day < 1 or Day > 365) [ AssertionProc "Error in calculation of Day! (Updates_Proc)" ]
set Date DateREP
if Day > 0 [ set DailyForagingPeriod_s Foraging_PeriodREP ]
UpdateFoodsourcesProc
UpdateSeasonalEventsProc
EmergenceNewQueensProc
let randomTimeToGetUp_s 1800 ; [s]
set ActiveBeesSortedList []
ask bees with [ (caste = "worker" or caste = "queen") and (activity != "hibernate") and stage =
"adult" ]
[
  set personalTime_s GetUpTime_s + random randomTimeToGetUp_s
  set activityList [ ]
  set ActiveBeesSortedList fput who ActiveBeesSortedList
]
set ActiveBeesSortedList sort-by [[personalTime_s] of bee ?1 < [personalTime_s] of bee ?2]
ActiveBeesSortedList
if Day = 365 and length CropRotationList > 1 \[ CropRotationProc \]
UpdateColoniesProc
UpdateColonyStoreBarsProc
CheckNumbersProc
end
```

; *****

to UpdateSeasonalEventsProc

; addresses seasonal events such as the species specific end of the season
ask Species

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

```
[
  let whoSpec who

  if Day = seasonStop
  [
    ask bees with [ speciesID = whoSpec and activity != "hibernate" ]
    [
      let memoNumber number
      if brood? [ ask colony colonyID [ set broodDeathEndSeason broodDeathEndSeason +
memoNumber ] ]
      DieProc "End of season"
    ]
  ]
]
if TotalHibernatingQueens = TotalQueens and (TotalEggs + TotalLarvae + TotalPupae = 0) and
TotalMales > 0; i.e. kill males in autumn if all queens are in hibernation and no brood is left
[
  ask bees with [ caste = "male" ][ DieProc "Males: all queens in hibernation!" ]
]
if Day = 1
[
  set PopulationSizeDay365List lput sum \[number\] of bees PopulationSizeDay365List
  with-local-randomness [ ask bees with [ caste = "queen" and activity = "hibernate" ] [ setxy 0 0 ] ]
; hibernating queens are moved to bottem left corner to distinguish this year's and last years queens
; with-local-randomness: to not change sequence of random numbers, results of "Version test",
2017-04-21
  let queensToKill TotalHibernatingQueens - MaxHibernatingQueens
  if queensToKill > 0
  [
    set TotalHibernatingQueensEverRemoved TotalHibernatingQueensEverRemoved +
queensToKill
    output-print "Reduced number of hibernating queens to no more than
MaxHibernatingQueens!"
  ]
  while [ queensToKill > 0 ]
  [
    ask one-of bees with [ caste = "queen" and activity = "hibernate" ] ;
hibernatingQueensToBeKilled of hibernating queens are killed on 1st January
    [
      set queensToKill queensToKill - number
      DieProc "max. number of hibernating queens"
    ]
  ]
]
ask foodsources
[
if CumulVisitsOnlyLastYear? = true
[
set cumulNectarVisits 0
set cumulPollenVisits 0
]
]
```

Twiston-Davies, Becher & Osborne. (2021). BEE-STEWARD: a research and decision support software for effective land management to promote bumblebee populations. *Methods in Ecology and Evolution*.

```
]
end
```

```
. *****
;
```

to-report WintermortalityProbREP

```
; Winter survival (survivalProb) is calculated from Beekman et al 1998 (Entomologia Experimentalis et Applicata 89: 207–214, 1998)
```

```
; Fig. 1B: survival prob. is calculated from proportion of survivors to survivors + non-survivors. Using the relative weight rather than the absolute weight, we fitted a sigmoid curve (survivalProb) to the left site only,
```

```
; as the low surv. prob. of heavy queens is an artefact of the treatment:
```

```
; "One would expect that queens with the highest weight will survive diapause. It is therefore surprising that the initial weight distribution of dead queens exceeds that of the surviving queens (Figure 1B and 1C).
```

```
; However, in 1993 the average initial weight of the queens was highest and in this period the most severe diapause regimes (6 or 8 months) were started. Since the majority of the queens that were given a treatment
```

```
; with a length of 6 or 8 months died, the initial weight distribution of dead queens exceeds that of the surviving queens."
```

```
let minWeightSpecies_mg [ devWeight_Q_PupationMin_mg ] of oneSpecies speciesID ; the min. weight of a queen for this species
```

```
let maxWeightSpecies_mg [ devWeight_Q_PupationMax_mg ] of oneSpecies speciesID ; the max. weight of a queen for this species
```

```
let myRelativeWeight (weight_mg - minWeightSpecies_mg) / (maxWeightSpecies_mg - minWeightSpecies_mg) ; the relative weight of this queen
```

```
if myRelativeWeight < 0 or myRelativeWeight > 1 [ AssertionProc "Wrong weight! (WintermortalityProbREP)" ]
```

```
let survivalProb 0.64 / (1 + e ^ (-22 * (myRelativeWeight - 0.32))) ; survival probability of B. terrestris, based on Beekman et al. 1998, with bees' weight relative to the min/max weights in this data set
```

```
report survivalProb
```

```
end
```

```
. *****
;
```