

CHANGES of BEE-STEWARD code in comparison to Bumble-BEEHAVE

extensions [profiler csv]

breed [foodsources foodsource]

breed [colonies colony]

breed [bees bee]

breed [species oneSpecies]

breed [badgers badger]

breed [signs sign]

breed [storebars storebar]

breed [deadCols deadCol]

[; from BEESCOUT](#)

[breed \[patchStatistics patchStatistic \]](#)

[; new for BEESTEWARD:](#)

[breed \[buttons button \]](#)

[breed \[buttonLabels buttonLabel \]](#)

[breed \[brushSigns brushSign \]](#)

[breed \[habitats habitat \]](#)

[habitats-own](#)

[

[flowerspecieslist](#)

[habitatColourID](#)

[colourRangeMin](#)

[colourRangeMax](#)

[habitatType](#)

[habitatSwitchedOn?](#)

]

bees-own

[

activity;"hibernate", "nestConstruction", "resting", "searching", "returningEmpty",
"returningUnhappyN", "returningUnhappyP", "nectarForaging", "collectNectar", "bringingNectar",
"expForagingN", "pollenForaging", "collectPollen", "bringingPollen", "expForagingP", "eggLaying",
"nursing"

activityList

adultAge

allelesList ; list with 1 (males) or 2 (females, dipl. males) alleles, if InbreedingEffects? true they
refer to the sex alleles with diploid males dying as adults

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*.

```
brood? ; true: brood, false: adult
broodAge
caste ; "undefined", "queen", "worker", "male"
colonyID
cropvolume_myl
cumulIncubationReceived_kJ
cumulTimeEgg_d
cumulTimeLarva_d
cumulTimePupa_d
currentFoodsource
emergingDate ; day of emergence from hibernation in ticks
expectation_NectarTrip_s
expectation_PollenTrip_s
glossaLength_mm
knownMasterpatchesNectarList ; lists the masterpatchID's of all nectar providing patches
('layergroups') sorted by the distance to the bee's colony
knownMasterpatchesPollenList ; similar for pollen
mated?
mtDNA
nectarLoadSquadron_kJ
nectarsourceToGoTo
number
personalTime_s
ploidy ; 1: haploid male, 2: diploid female (or diploid male)
pollenForager?
pollenLoadSquadron_g
pollenPellets_g
pollenSourceToGoTo
speciesID
speciesName
spermathecaList
stage ; egg, larva, pupa, adult
thEggLaying
thForagingNectar
thForagingPollen
thNursing
weight_mg
]

;.....
```

colonies-own ; colony and species specific variables ; COPY ANY CHANGES TO DEADCOLS-OWN
VARIABLES

```
[
  allAdultActiveQueens
  allAdultMales
  allAdultQueens
  allAdults
  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*.

allEggs
allLarvae
allPatchesInRangeList
allPupae
allSourcesInFlowerAndRangeList ; all foodsources (including 'layers') within the foraging range that provide nectar a/o pollen, updated, whenever a foodsource has a start or a stop day
broodDeathBadger
broodDeathEndSeason
broodDeathsCP
broodDeathsEnergyStores
broodDeathsNoAdults
cohortBased?
colonyAge
colonyFoundationDay
colonySize
colonyWeight_mg
competitionPointDate
eggDeathsIncubation
energyNeedToday_kJ
energyStore_kJ
eusocialPhaseDate ; emergence of the the first worker (Duchateau & Velthuis 1988)
idealEnergyStore_kJ
idealPollenStore_g
larvaDeathsIncubation
larvaDeathsWeight
larvaWorkerRatio
masterpatchesInRangeList ; all masterpatches within foraging range, determined only once, when colony is created
masterpatchesWithNectarlayersInFlowerAndRangeList ; all masterpatches within the foraging range where at least one layer provides nectar today (in principle, i.e. it might become depleted during a day); updated, whenever a foodsource has a start or a stop day
masterpatchesWithPollenlayersInFlowerAndRangeList ; ditto for pollen
nectarInFlowerAndRangeList
pollenInFlowerAndRangeList
pollenNeedLarvaeToday_g
pollenStore_g
pupaDeathsIncubation
queenProduction?
queenProductionDate ; the first date when diploid larvae eggs were laid that can develop into queens
queenright?
speciesIDcolony
[speciesNameColony](#)
stimEggLaying
stimNectarForaging
stimNursing
stimPollenForaging
summedIncubationToday_kJ
switchPointDate
totalAdultsProduced

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*.

```
totalEggsProduced
totalLarvaeProduced
totalMalesProduced
totalPupaeProduced
totalQueensProduced
totalWorkersProduced
]
```

```
deadCols-own
[
  colonyDeathDay
  allAdultActiveQueens
  allAdultMales
  allAdultQueens
  allAdults
  allAdultWorkers
  allEggs
  allLarvae
  allPatchesInRangeList
  allPupae
  allSourcesInFlowerAndRangeList ; all foodsources (including 'layers') within the foraging range that
  provide nectar a/o pollen, updated, whenever a foodsource has a start or a stop day
  broodDeathBadger
  broodDeathEndSeason
  broodDeathsCP
  broodDeathsEnergyStores
  broodDeathsNoAdults
  cohortBased?
  colonyAge
  colonyFoundationDay
  colonySize
  colonyWeight_mg
  competitionPointDate
  eggDeathsIncubation
  energyNeedToday_kJ
  energyStore_kJ
  eusocialPhaseDate ; emergence of the the first worker (Duchateau & Velthuis 1988)
  idealEnergyStore_kJ
  idealPollenStore_g
  larvaDeathsIncubation
  larvaDeathsWeight
  larvaWorkerRatio
  masterpatchesInRangeList ; all masterpatches within foraging range, determined only once, when
  colony is created
  masterpatchesWithNectarlayersInFlowerAndRangeList ; all masterpatches within the foraging
  range where at least one layer provides nectar today (in principle, i.e. it might become depleted
  during a day); updated, whenever a foodsource has a start or a stop day
  masterpatchesWithPollenlayersInFlowerAndRangeList ; ditto for pollen
  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*.

```
pollenInFlowerAndRangeList
pollenNeedLarvaeToday_g
pollenStore_g
pupaDeathsIncubation
queenProduction?
queenProductionDate ; the first date when diploid larvae eggs were laid that can develop into
queens
queenright?
speciesIDcolony
stimEgglaying
stimNectarForaging
stimNursing
stimPollenForaging
summedIncubationToday_kJ
switchPointDate
totalAdultsProduced
totalEggsProduced
totalLarvaeProduced
totalMalesProduced ; renamed from allMalesProduced
totalPupaeProduced
totalQueensProduced ; renamed from allQueensProduced
totalWorkersProduced
]
```

;.....

```
species-own
[
  batchsize
  chanceFindNest ;chance of queen finding a nest site per day
  dev_larvalAge_QueenDetermination_d ; day of larval development when female bee either
develops into worker or queen (i.e. now independent of age of hatching from the egg!)
  dev_Q_DeterminationWeight_mg
  devAge_Q_EmergingMax_d
  devAge_Q_EmergingMin_d
  devAge_Q_PupationMax_d
  devAge_Q_PupationMin_d
  devAgeEmergingMax_d
  devAgeEmergingMin_d
  devAgeHatchingMax_d
  devAgeHatchingMin_d
  devAgePupationMax_d
  devAgePupationMin_d
  devIncubation_Q_EmergingTH_kJ
  devIncubation_Q_PupationTH_kJ
  devIncubationEmergingTH_kJ
  devIncubationHatchingTH_kJ
  devIncubationPupationTH_kJ
  devQuotaIncubationToday_kJ
  devWeight_Q_PupationMax_mg
```

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*.

```
devWeight_Q_PupationMin_mg
devWeightEgg_mg
devWeightPupationMax_mg
devWeightPupationMin_mg
emergingDay_mean
emergingDay_sd
flightCosts_kJ/m/mg
flightVelocity_m/s
growthFactor
maxLifespanWorkers
minPollenStore_g
minToMaxFactor
name
nestHabitatsList
nestSiteArea
nestsiteFoodsourceList ; list of foodSources suitable for species nest sites
pollenToBodymassFactor
proboscis_max_mm
proboscis_min_mm
searchLength_m
seasonStop
specMax_cropVolume_myl ; species maximum size of crop (regardless of weight)
specMax_pollenPellets_g ; species maximum size of pellets (regardless of weight)
timeUnloading
]
```

.....

```
foodsources-own
[
  area_sqm
  colorMemo
  corollaDepth_mm ; (average) length of the corolla tubes
  cumulNectarVisits
  cumulPollenVisits
  flowerSpecies_relativeAbundanceList
  flowerSpeciesList
  id_Beescout
  interFlowerTime_s
  layersInPatchList ; lists who of all foodsources what belong to the same flower patch ('layergroup'),
saved for masterpatches only
  masterpatch? ; boolean, true for first layer in a 'layergroup'
  masterpatchID ; the ID of the masterpatch of this layers' 'layergroup'
  nectar_myl
  nectarConcentration_mol/l
  nectarFlowerVolume_myl
  nectarMax_myl
  patchInfo ; short string with additional information for the user
  patchType
```

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.

```
pollen_g
pollenMax_g
proteinPollenProp
radius_m
startDay ; first day of flowering period
stopDay ; first day AFTER the flowering period
farmland?
stewardshipSpeciesList
perimeter\_m
]
```

```
;.....
```

```
patches-own
[
  nColonies
  nMalesProduced
  nQueensProduced
  pcolorSave
]
```

```
;.....
```

```
; former BEESCOUT variables:
flowerPatchID
firstPatchOfFlowerpatch
mapDisplay
originalColor
patchColor
habitatTypePatch
satelliteColor
visits
```

```
]
```

```
storebars-own
[
  maxSize
  store
  storeColonyID
]
```

```
;.....
```

```
patchStatistics-own ; former BEESCOUT variables
[
```

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.

```
_areaPx  
_areaSqm  
_patchType  
_flowerSpeciesList  
_patchInfo  
_perimeter_m  
]
```

```
buttons-own  
[  
_myLabel  
_myCommand  
_on?  
_exitButton?  
_headerButton?  
]
```

globals

```
[  
;; GLOBAL PARAMETERS DEFINED ON INTERFACE  
;;; GenericPlot1 "Number of colonies for different species"  
;;; Panel "Maps and Settings"  
;;; Period "till 31st December"  
  
;; HIDDEN BEHIND "Output" WINDOW:  
;;; BrushSize  
;;; BS_Scaling_m/NLpatches  
;;; CircularBrush?  
;;; GIS_ncols  
;;; GIS_NoDataValue  
;;; GIS_nrows  
;;; GIS_xllcorner  
;;; GIS_yllcorner  
;;; MyFarmlandPatches  
;;; MyMap  
;;; MyParametersFile  
;;; MySavedMap1  
;;; MySavedMap2  
;;; MySavedMap3  
;;; MySavedMap4  
;;; MySavedMap5  
;;; MySavedMap6  
;;; ProjectsOwnParameterFile?  
;;; RAND_SEED  
;;; Scaling_NLpatches/m  
;;; StewardshipOption  
  
_AbundanceBoost  
_ActiveBee
```


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.

- [_ ActiveBeesSortedList](#)
- [_ AllParametersList](#)
- [_ AssertionMessage](#) ; allow message to be recorded out by RNetLogo
- [_ AssertionViolated](#)
- [_ BackgroundColor](#)
- [_ Backgroundcolour](#)
- [_ BackgroundImage](#)
-
- [_ BehaviourSpaceParameters](#)
- [_ BehavSpaceParameterValuesList](#)
- [_ BehavSpaceCurrentValue](#)
- [_ Black_TH](#)
- [_ BorderColor](#)
- [_ BottomBorder](#)
- [_ BrushArea_ha](#)
- [_ BS_ScaleDistance_m](#)
- [_ BumbleBeehave_Inputfile](#)
- [_ Button1Monitor](#)
- [_ Button2Monitor](#)
- [_ Button3Monitor](#)
- [_ Button4Monitor](#)
- [_ Button5Monitor](#)
- [_ Button6Monitor](#)
- [_ Button7Monitor](#)
- [_ CallItaDay_s](#)
- [_ ChosenPatchtypeVariable](#)
- [_ CohortSymbolSize](#)
- [_ COLONIES IBM](#)
- [_ ColonyDeathsEndSeason](#) ; count of colonies removed due to the season ending
- [_ ColonyDeathsNoBees](#) ; count of colonies removed due to no adult bees
- [_ ColonyDensity_km2](#)
- [_ ColonySymbolSize](#)
- [_ ColorIBM](#)
- [_ ColourCodeList](#)
- [_ CropRotationList](#)
- [_ CropRotationListAsString](#)
- [_ CSS_OptionsList](#)
- [_ CSS_TotalAreaAll_ha](#)
- [_ CSS_TotalAreaLegume_ha](#)
- [_ CSS_TotalAreaMargin_ha](#)
- [_ CSS_TotalAreaPlot_ha](#)
- [_ CumulVisitsOnlyLastYear?](#)
- [_ DailyForagingPeriod_s](#)
- [_ DailySwitchProbability](#)
- [_ Date](#)
- [_ Day](#)
- [_ Daytime_s](#)
- [_ EnergyFactorOnFlower](#)
- [_ EnergyHoney_kJ/ml](#)
- [_ EnergyRequiredForPollenAssimilation_kJ_per_g](#)

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.

- [_ EnergySucrose_kJ/mymol](#)
- [_ FlowerspeciesFile](#)
- [_ FlowerSpeciesList Legume](#)
- [_ FlowerSpeciesList Margin](#)
- [_ FlowerSpeciesList Plot](#)
- [_ FoodSourceLimit](#)
- [_ FoodsourcesInFlowerUpdate? ; true on days when at least 1 foodsource starts or stops flowering](#)
- [_ FoodsourceSymbolSizeFactor](#)
- [_ ForagingFileList](#)
- [_ ForagingMortalityFactor](#)
- [_ ForagingMortalityModel](#)
- [_ ForagingRangeMax_m](#)
- [_ Gap_s](#)
- [_ GenericInputNumber](#)

- [_ ;GenericPlot2](#)
- [_ ;GenericPlot3](#)
- [_ ;GenericPlot4](#)
- [_ ;GenericPlot5](#)

- [_ GenericRunCommandValue](#)
- [_ GetUpTime_s ; beginning of the working day](#)
- [_ Gridsize](#)
- [_ HabitatDataCSV](#)
- [_ HabitatsFile ; HabitatsInput RENAMED](#)
- [_ FoodsourcesFile](#)
- [_ InspectTurtle](#)
- [_ KeepDeadColonies?](#)
- [_ Lambda_detectProb](#)
- [_ LarvaWorkerRatioTH](#)
- [_ MapAreaIncluded](#)
- [_ MasterSizeFactor](#)
- [_ MaxForagingRange_m](#)
- [_ MaxHibernatingQueens](#)
- [_ MaxLifespanMales](#)
- [_ MaxPatchRadius_m](#)
- [_ MeanAdultWeight_mg](#)
- [_ MeanQueenWeight_mg](#)
- [_ MeanWorkerWeight_mg](#)
- [_ MergeHedges?](#)
- [_ MetabolicRateFlight_W/kg](#)
- [_ MinFoodSourceSymbolSize](#)
- [_ MortalityForager_per_s](#)
- [_ MinSizeFoodSources?](#)
- [_ MortalityAdultsBackground_daily](#)
- [_ MortalityForager_per_s](#)
- [_ MyValue](#)
- [_ N_Badgers](#)
- [_ N_ForeignAlleles](#)
- [_ N_Psithyrus](#)

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.

- [_NectarAvailableTotalToday_l](#)
- [_NestSearchTime_h](#)
- [_NotSetHigh](#)
- [_NotSetLow](#)
- [Npatches](#)
- [OutputWordResult](#)
- [PatchColoursList](#)
- [Patchtype Legume](#)
- [Patchtype Margin](#)
- [Patchtype Plot](#)
- [_PollenAvailableTotalToday_kg](#)
- [_QueenDestinedEggsBeforeSP_d](#)
- [_QueensProducingColoniesList](#)
- [_QueenSymbolsize](#)
- ~~[_Scaling_NLpatches/m](#)~~
- [_RemoveEmptyFoodSources?](#)
- [_Repetitions](#)
- [_RightBorder](#)
- [_SatelliteFile](#)
- [_SelectedFieldsList](#)
- [_SetColour](#)
- [_SexLocus?](#)
- [_ShowCohorts?](#)
- [_ShowDeadCols?](#)
- [_ShowFoodsources?](#)
- [_ShowGrid?](#)
- [_ShowInspectedColony?](#)
- [_ShowMasterpatchesOnly?](#)
- [_ShowNests?](#)
- [_ShowPlots?](#)
- [_ShowQueens?](#)
- [_ShowSearchingQueens?](#)
- [_ShowWeather?](#)
- [_BeespeciesFile ; SpeciesFilename RENAMED](#)
- [_SpeciesList](#)
- [_StepWidth ; for drawing colony cohorts](#)
- [_StopExtinct?](#)
- [_Sunrise_s](#)
- [_SwitchOn](#)
- [_TextMap](#)
- [_TopBorder](#)
- [_TotalActiveBees](#)
- [_TotalAdultMales](#)
- [_TotalAdultMalesEverProduced](#)
- [_TotalAdultQueens](#)
- [_TotalAdultQueensEverProduced](#)
- [_TotalAdults](#)
- [_TotalAdultsEverProduced](#)
- [_TotalAdultWorkers](#)
- [_TotalBeeAgents](#)

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*.

```

_ TotalBeesEverDied
_ TotalBeesEverProduced
_ TotalColonies
_ TotalColoniesEverProduced ; number of colonies formed during a model run
_ TotalEggs
_ TotalFoodSources
_ TotalForagingTripsToday
_ TotalHibernatingQueens
_ TotalHibernatingQueensEverRemoved
_ TotalIBMColonies
_ TotalLarvae
_ TotalMales
_ TotalMapArea km2
_ TotalMatedQueens
_ TotalPupae
_ TotalQueens
_ ;;TotalTrips
_ TotalUnmatedQueens
}

```

UnlimitedMales?
 UserHabitatSelected
 Weather
 White_TH
 WinterMortality?

BeeSpeciesInitialQueensList
 BeeSpeciesInitialQueensListAsString
 Report_name
 BeeSpeciesDefinedList
 InitialQueensModifierBehaviorSpace
 PopulationSizeDay365List
 Survived?
 TimeToExtinction

```

]

; =====

, *****
,

```

to ParametersProc

; this procedure sets the GLOBAL parameters of the model

```

QueensInitialProc
set SpeciesList {}
-if B_terrestris > 0 [ set SpeciesList fput "B_terrestris" SpeciesList]
-if B_pascuorum > 0 [ set SpeciesList fput "B_pascuorum" SpeciesList]
-if B_lapidarius > 0 [ set SpeciesList fput "B_lapidarius" SpeciesList]

```

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 B_hortorum > 0 [set SpeciesList fput "B_hortorum" SpeciesList]
-if B_hypnorum > 0 [set SpeciesList fput "B_hypnorum" SpeciesList]
-if B_pratorum > 0 [set SpeciesList fput "B_pratorum" SpeciesList]
-if N_Psithyrus > 0 [set SpeciesList fput "Psithyrus" SpeciesList]
```

```
set AssertionViolated false
set CallItaDay_s 24 * 3600 ; [s]
set CohortSymbolSize 0.75 * MasterSizeFactor
set ColonySymbolSize 9 * MasterSizeFactor
set ColorIBM 24 ; (24 = dark orange)
set DailySwitchProbability 0.13 ; derived from Duchateau & Velthuis 1988 (50% of the (early
switching) colonies switch within ca. 2*2.4d, i.e. 13% per day)
set EnergyFactorOnFlower 0.3 ; for honeybees: Kacelnik et al 1986 (BES:19): 0.3 (rough estimation,
based on Nunez 1982)
set EnergyHoney_kJ/ml 22.67 ; 22.67 [kJ/ml] ; from B. vosnesenskii, Allen et al 1978 (J KANSAS
ENTOMOLOGICAL SOCIETY; 51(3), 1978, pp. 329-342): Tab.2:
; 195ml honey contains 260g sugar; Energy Sucrose ca. 17 kJ/g; 260g * 17kJ/g =
4420 kJ energy stored in honey
; 4420kJ / 195ml = 22.67 kJ/ml (or 1 kJ = 0.04412 ml)
set EnergyRequiredForPollenAssimilation_kJ_per_g 6.2 ;; Hrassnig, Crailsheim 2005 (honeybee
larvae): consumes ca. 156.25 mg pollen (125-187.5 mg, Tab 1.) and 59.4 mg carbohydrates (Tab. 1,
from Rortais et al 2005).
;; Energy carbohydrates ca. 16.3 kJ/g (3.89kcal/g * 4.19 = 16.3 kJ/g
(http://ndb.nal.usda.gov)),
;; hence: energy from carbohydrates = 59.4mg * 16.3 kJ/g = 968.22 kJ
to assimilate 156.25 mg pollen or 6.1966 kJ for 1 mg pollen
set EnergySucrose_kJ/mymol 0.00582 ; 0.00582 [kJ/micromol] 342.3 g/mol from BEEHAVE
set FoodsourcesInFlowerUpdate? false
set FoodsourceSymbolSizeFactor 1.5 * MasterSizeFactor
set ForagingRangeMax_m 758 ; 758m: Knight et al. 2005 ; 1500m: Osborne et al. 2008,
set GetUpTime_s 1 ; 1 (i.e: 0:00:01 AM)
set LarvaWorkerRatioTH 3 ; Duchateau & Velthuis 1988
set MaxLifespanMales 30 ; 30d of adult age; Duchateau & Marien 1995 Ins. Soc. 42:255-266
(1995): 30.48+-10.23; however: bees were kept in flight-cages hence most likely overestimating life
span
set MetabolicRateFlight_W/kg 488.6 ; Wolf et al. 1999 (Tab. 1, Open air: 488.6+-226.7 W/kg)
set MinFoodSourceSymbolSize 2.5 * MasterSizeFactor
if ForagingMortalityModel = "high" [ set MortalityForager_per_s 1.0E-05 ] ; (BEEHAVE VALUE:
0.00001, from Visscher&Dukas 1997 (Mort 0.036 per hour foraging)
if ForagingMortalityModel = "intermediate" [ set MortalityForager_per_s 2.14E-06 ] ; (Schmid-
Hempel & Heeb 1991: mortality (B. lucorum) 30-40% per week (=>35%), survival rate per week: 0.65
; assuming 8hrs foraging per day: 7 * 8 * 3600 = 201600 seconds, survival rate/s =
0.65^(1/201600) => mortality rate/s 2.14E-06
if ForagingMortalityModel = "low" [ set MortalityForager_per_s 2.75E-07 ] ; Stelzer et al. 2010 (B.
terrestris) (doi:10.1111/j.1469-7998.2010.00709.x), Tab. 1 (from mean of loss rate %/h)
set MortalityAdultsBackground_daily 0 ; Plowright & Jay 1968: negligible adult mortality in captive
colonies (B. ternarius)
set N_ForeignAlleles 24 ; (24) "The number of sex alleles in this population is estimated to be at
least 24" Duchateau et al. 1994 Entomol. exp. appl. 71: 263-269, 1994.
set NestSearchTime_h 6
```

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 NotSetHigh 9999999999999999
set NotSetLow -1 * NotSetHigh
set QueenDestinedEggsBeforeSP_d 5 ; First queen destined egg is laid ca. 5d before SP (Duchateau & Velthuis 1988 , Fig. 4: Queen destined eggs laid 5d before SP (or later))
set QueensProducingColoniesList []
set QueenSymbolSize 2 * MasterSizeFactor
set StepWidth 0.5 * MasterSizeFactor
set Sunrise_s 8 * 3600 ; 8 * 3600 = 8:00 a.m.
```

```
;; NEW FOR Bee-Steward:
set FoodsourcesFile (word remove ".png" MyMap " Foodsources.txt")
set BackgroundColor 5
set SelectedFieldsList []
set CSS_OptionsList [ "margin" "legume" "plot" ] ;;; "blueberryPlot" ] ; CAUTION: Netlogo
'member?'-command is CASE SENSITIVE!!!
set CropRotationList ListFromStringREP CropRotationListAsString

set PopulationSizeDay365List [] ; for output, list is populated on each 1st Jan (starting at ticks = 1) in
UpdateSeasonalEventsProc
set Survived? 1

;clear-output
foreach AllParametersList
[
  let command (word "set GenericRunCommandValue " ?)
  run command
]
end
```

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

to QueensInitialProc

```
; Bees species (e.g. B terrestris) are no longer global variables defined by GUI Input. Instead, the
initial number of queens for
; each species is defined in the variable BeeSpeciesInitialQueensListAsString, read in from the
parameter file
; This procedure uses the information in the string variable BeeSpeciesInitialQueensListAsString to
create the list BeeSpeciesInitialQueensList
;; BeeSpeciesInitialQueensListAsString: e.g. "B terrestris 517 O rufa 100"
```

```
set SpeciesList []
set BeeSpeciesInitialQueensList []
let beeSpeciesInitialQueensList preliminary ListFromStringREP BeeSpeciesInitialQueensListAsString
; format of prelim. list e.g. ["B hortorum" "100" "B terrestris" "500"] - whereas the format in the
end should look like ["B hortorum" 100] ["B terrestris" 500]]
let newSpecies true ; keeps track whether the next item is a species name or a number
let nextEntry []
```

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*.

```

foreach beeSpeciesInitialQueensList preliminary ;; e.g. e.g. ["B_hortorum" "100" "B_terrestris"
"500"]
[
  ifelse newSpecies = true ; i.e. ? = e.g. "B_hortorum"
  [
    set nextEntry lput ? nextEntry ;; => ["B_hortorum"]
    set newSpecies false
  ]
  [ ;; ELSE ; i.e. ? = e.g. "100"
    set nextEntry lput read-from-string ? nextEntry ;; => ["B_hortorum" 100] (read-from-string to
remove the quotation marks)
    set BeeSpeciesInitialQueensList lput nextEntry BeeSpeciesInitialQueensList
    set newSpecies true
    set nextEntry []
  ]
]
foreach BeeSpeciesInitialQueensList ; e.g. ["B_hortorum" 100] ["B_terrestris" 500]]
[ set SpeciesList lput item 0 ? SpeciesList ]
end

```

, *****

to CreateSpeciesProc

```

ifelse ( file-exists? SpeciesFilename ) ; ***MB*** 2018-04-13
[
  let speciesDataCSV csv:from-file BeespeciesFileSpeciesFilename ; a csv input file is loaded and saved
  let header item 0 speciesDataCSV ; first line of the input file is the header
  set BeeSpeciesDefinedList []
  foreach speciesDataCSV ; goes through all lines in ordered way
  [
    set BeeSpeciesDefinedList lput item 0 ? BeeSpeciesDefinedList
    if member? item 0 ? SpeciesList ; if the species (i.e.first entry) of the current row is member of the
SpeciesList (i.e. the list with those bee species added to the simulation, which was created in
ParametersProc)
    [
      create-species 1
      [
        set name item (position "name" header) ? ; checks in which column of the input data the
species are listed and uses the value of the current row
        set maxLifespanWorkers item (position "maxLifespanWorkers" header) ?
        set emergingDay_mean item (position "emergingDay_mean" header) ?
        set emergingDay_sd round (item (position "emergingDay_sd" header) ?)
        set batchsize item (position "batchsize" header) ?
        set flightVelocity_m/s item (position "flightVelocity_m/s" header) ?
        set flightCosts_kJ/m/mg MetabolicRateFlight_W/kg / flightVelocity_m/s / (1000 * 1000 * 1000)
; W/kg = J/s/kg; div. by speed => J/m/kg i.e. 0.001kJ/s/(1000000*mg)
        set searchLength_m item (position "searchLength_m" header) ?
        set seasonStop item (position "seasonStop" 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*.

```

set timeUnloading item (position "timeUnloading" header) ?
set specMax_cropVolume_myl item (position "specMax_cropVolume_myl" header) ?
set specMax_pollenPellets_g item (position "specMax_pollenPellets_g" header) ?
set nestHabitatsList []
set nestHabitatsList read-from-string item (position "nestHabitatsList" header) ?
set minToMaxFactor item (position "minToMaxFactor" header) ?
set devAgeHatchingMin_d item (position "devAgeHatchingMin_d" header) ?
set devAgePupationMin_d item (position "devAgePupationMin_d" header) ?
set devAgeEmergingMin_d item (position "devAgeEmergingMin_d" header) ?
set devWeightEgg_mg item (position "devWeightEgg_mg" header) ?
set devWeightPupationMin_mg item (position "devWeightPupationMin_mg" header) ?
set devWeightPupationMax_mg item (position "devWeightPupationMax_mg" header) ?
set pollenToBodymassFactor item (position "pollenToBodymassFactor" header) ?
set dev_Q_DeterminationWeight_mg item (position "dev_Q_DeterminationWeight_mg"
header) ?
set devAge_Q_PupationMin_d item (position "devAge_Q_PupationMin_d" header) ?
set devWeight_Q_PupationMin_mg item (position "devWeight_Q_PupationMin_mg" header) ?
set devWeight_Q_PupationMax_mg item (position "devWeight_Q_PupationMax_mg" header) ?
set devAge_Q_EmergingMin_d item (position "devAge_Q_EmergingMin_d" header) ?
set growthFactor item (position "growthFactor" header) ?
set proboscis_min_mm item (position "proboscis_min_mm" header) ?
set proboscis_max_mm item (position "proboscis_max_mm" header) ?
set chanceFindNest item (position "dailyNestSiteChance" header) ? ;dailychance for queen
finding nest site
set devQuotaIncubationToday_kJ 10 / (1.5 * batchsize) ; 10 kJ: Silvola 1984: Queen (B.
terrestris) spends ca. 10kJ/day for incubation; during this time it incubates about 1.5 batches of
brood
set devAgeHatchingMax_d devAgeHatchingMin_d * minToMaxFactor
set devAgePupationMax_d devAgePupationMin_d * minToMaxFactor - devAgeHatchingMin_d
set devAgeEmergingMax_d devAgeEmergingMin_d * minToMaxFactor - devAgePupationMin_d
set devIncubationHatchingTH_kJ devQuotaIncubationToday_kJ * devAgeHatchingMin_d
set devIncubationPupationTH_kJ devQuotaIncubationToday_kJ * devAgePupationMin_d
set devIncubationEmergingTH_kJ devQuotaIncubationToday_kJ * devAgeEmergingMin_d
set dev_larvalAge_QueenDetermination_d 3 ; (3d) (Roeseler 1970 Z. Naturforsch. 25 b, 543—
548: within the first 3.5 days of larval development
set devAge_Q_PupationMax_d devAge_Q_PupationMin_d * minToMaxFactor -
devAgeHatchingMin_d
set devAge_Q_EmergingMax_d devAge_Q_EmergingMin_d * minToMaxFactor -
devAge_Q_PupationMin_d
set devIncubation_Q_PupationTH_kJ devQuotaIncubationToday_kJ *
devAge_Q_PupationMin_d
set devIncubation_Q_EmergingTH_kJ devQuotaIncubationToday_kJ *
devAge_Q_EmergingMin_d ; * Incubation_Q_Factor

;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:

```


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 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 foodsources. No colonies will form")]

—}
—}
—}
}
; if SpeciesFilename doesn't exist
[user-message {word "The file SpeciesFilename ("SpeciesFilename") doesn't exist in the current
directory. Check file names!"}]
—]
—]
—]
set BeeSpeciesDefinedList but-first BeeSpeciesDefinedList ; first entry was the header ("species")
end

```

```

; *****
;

```

; Reporter to find foodSource for queens that have found a nest site.

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
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 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

```

, *****

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

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 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 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
    ]
  ]
]

```

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 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
```

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

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" ]
  [
    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..
```

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 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"
set color yellow
]
```

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 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 ; 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:
      ;;; OLD: id patchType patchColour xcor ycor size_sqm quantityPollen_g quantityNectar_l
concentration startDay stopDay corollaLength_mm nectarFlowerVolume_myl interFlowerTime_s
patchInfo
      ; id patchType patchColour xcor ycor size\_sqm flowerSpeciesList perimeter\_m info

```

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 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_myl AbundanceBoost * file-read * 1000 * 1000 ; [quantityNectar_l: l * 1000 = ml; ml
* 1000 = myl]
set nectarMax_myl nectar_myl
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_myl 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
set stewardshipSpeciesList []
if member? (word " " who " ") MyFarmlandPatches [ set farmland? true ] ; CAUTION: Netlogo
'member?'-command is CASE SENSITIVE!!!
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 ]
if MapAreaIncluded = "bottom right 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 = "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 (""FoodsourcesFile
INPUT_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 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.

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)

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 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 removedfoodsources with [ patchType =
"Hedgerow" ]
```

```
[
```

```
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 ?
```

```
let pollenToBeAdded_g [pollenMax_g] of foodsource ?
```

```
ask foodsource masterHedgeID
```

```
[
```

```
set area_sqm area_sqm + areaToBeAdded_sqm
```

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 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 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

; 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.

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 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 ; = true
```

```
    [ 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
```

```
    ; 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
```

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*.

```

__ ; 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 ;; 1000 *
(item (position "nectar_ml/flower" header) myDataLine)
__ set nectarConcentration_mol/l (item (position "concentration_mol/l" header) myDataLine) ;
[ mol/l ] "position" determines the column with the relevant data
__ 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 ] ; for some
reason, myCSOarea m2 can still be a string here!
    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 ] ; for some
reason, myCSOarea m2 can still be a string here!

```

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 CSS_TotalAreaPlot ha CSS_TotalAreaPlot ha + myCSOarea_m2 / 10000
set CSS_TotalAreaAll ha CSS_TotalAreaAll ha + myCSOarea_m2 / 10000
_]

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 ] ; for some
reason, myCSOarea_m2 can still be a string here!
set CSS_TotalAreaLegume ha CSS_TotalAreaLegume ha + myCSOarea_m2 / 10000
set CSS_TotalAreaAll ha CSS_TotalAreaAll ha + myCSOarea_m2 / 10000
_]
_]
end
```

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

to ClearProc

```
clear-all
stop-inspecting-dead-agents
reset-ticks
end
```

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

to Setup

```
ClearProc
SetupWithoutClearingProc
end
```

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

to SetupBehaviorSpace

; only called when BehaviorSpace is used!

; 1.) When running a BehaviorSpace experiment, several cores might try to access and modify the parameter file at the same time, which causes errors. To avoid this, each run creates its own (temporary) parameter file, which is deleted at the end of the setup.

; 2.) In contrast to Bumble-BEEHAVE, most parameters of the model are no longer defined via input options on the GUI but read in. When using BehaviorSpace, this causes problems, because: either

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*.

; parameters are read in during Setup - which overwrites the values set in BehaviorSpace or this isn't done, which sets all parameters (except those defined in BehaviorSpace) to 0.

; This procedure solves this problem by (locally) saving the parameters (and their values) set in BehaviorSpace and then - after reading in the default values of all parameters, resetting the BehaviorSpace parameters to the previously saved values.

; This is possible, as "clear-all" doesn't re-set local variables.

; (However, the "run" command only works on global variables, so "BehavSpaceParameterValuesList" and "BehavSpaceCurrentValue" both have to be defined as global variables. "behavSpaceParameterVALUESListMemo" saves "BehavSpaceParameterValuesList"

; as a local variable (i.e. after Setup/ clear-all, BehavSpaceParameterValuesList will be 0 but behavSpaceParameterVALUESListMemo will still have the correct values)

; Also note: BehaviourSpaceParameters is a string (as lists cannot be defined in BehaviorSpace) and needs to be translated into a list first!)

let temporaryParameterFilename (word " ParameterFileTemporary " behaviorspace-run-number ".csv"); ; name of the temporary parameter file

set ProjectsOwnParameterFile? false ; this needs to be false, otherwise, the parameter file name automatically derived from the name of the map, will be used

set BehavSpaceParameterValuesList [] ; this variable is a list!

; first, save BehaviourSpaceParameters as a list:

let behaviourSpaceParametersListMemo []

let continue? true

if member? " " BehaviourSpaceParameters = false [set continue? false] ; i.e. if only one parameter (or none at all) is listed in BehaviourSpaceParameters

while [continue? = true]

└

let nextBlankPosition position " " BehaviourSpaceParameters

let nextItem substring BehaviourSpaceParameters 0 nextBlankPosition ; the first parameter in the string (incl. first character but not the blank)

set BehaviourSpaceParameters substring BehaviourSpaceParameters (nextBlankPosition + 1) length BehaviourSpaceParameters ; removes first parameter in the list as well as the first blank

set behaviourSpaceParametersListMemo lput nextItem behaviourSpaceParametersListMemo

if member? " " BehaviourSpaceParameters = false [set continue? false]

└

if BehaviourSpaceParameters != ""

└

set behaviourSpaceParametersListMemo lput BehaviourSpaceParameters

behaviourSpaceParametersListMemo ; BehaviourSpaceParameters here represents the last parameter, which needs to be added to the list

└

; save the current values of the parameters set in BehaviorSpace in the new list

BehavSpaceParameterValuesList

if length BehaviourSpaceParameters > 0

└

foreach behaviourSpaceParametersListMemo

└

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 nextCommand (word "set BehavSpaceCurrentValue " ? ) ; extra command here (replacing  
BehavSpaceCurrentValue in the next command by "?" creates an error)  
run nextCommand  
set nextCommand (word "set BehavSpaceParameterValuesList lput " BehavSpaceCurrentValue "  
BehavSpaceParameterValuesList") ; "run" cannot access local variable!  
run nextCommand  
] ]
```

```
let behavSpaceParameterVALUESListMemo BehavSpaceParameterValuesList ; the global  
BehavSpaceParameterValuesList is saved as local variable to survive clear-all!  
let beeSpeciesInitialQueensListAsStringMEMO BeeSpeciesInitialQueensListAsString  
let cropRotationListAsStringMEMO CropRotationListAsString
```

```
; set the parameters file to the default, " SYSTEM Parameters.csv" and read it in!  
set MyParametersFile " SYSTEM Parameters.csv"  
ReadAllParametersProc ; the new parameter settings are loaded  
Setup  
set ProjectsOwnParameterFile? false
```

```
; all parameters are set to their default values now - those defined in BehaviorSpace need to be re-  
set:
```

```
set StopExtinct? false ; make sure each run completes all time steps! (to set it true, list it in  
BehaviorSpace!)  
if behaviourSpaceParametersListMemo != [] ; i.e. only if there is actually a parameter defined that  
needs to be changed
```

```
[  
let i 0  
foreach behaviourSpaceParametersListMemo  
[  
let newValue item i behavSpaceParameterVALUESListMemo  
let nextCommand (word "set " ? " " newValue)  
run nextCommand  
set i i + 1  
]  
]
```

```
set BeeSpeciesInitialQueensListAsString beeSpeciesInitialQueensListAsStringMEMO  
set CropRotationListAsString cropRotationListAsStringMEMO  
if CropRotationListAsString = 0 [set CropRotationListAsString "" ] ; if using more than 1 core,  
CropRotationListAsString is 0 (unless it is listed in BehavSpace), causing an error ("MEMBER?  
expected input to be a string or list or agentset but got the number 0 instead.")  
set MyParametersFile temporaryParameterFilename
```

```
; all parameters are now set to their correct value. Save these in the parameter file and continue  
with Setup (and run):
```

```
SaveLoadSettingsProc "Save!" ""  
Setup ; now setup again with the correct parameterisation!  
if file-exists? temporaryParameterFilename [ file-delete temporaryParameterFilename ] ; and finally  
delete the temporary parameter file!  
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 SetupWithoutClearingProc

~~clear all~~

~~stop inspecting dead agents~~

~~reset ticks~~

if RAND_SEED != 0 [random-seed RAND_SEED]

;; OR ALTERNATIVELY: if RAND_SEED != 0 [random-seed RAND_SEED + 10000 * behaviorspace-run-number]

if (MyMap = " SYSTEM Example Farm.png" and MyParametersFile = " SYSTEM Example Farm Parameters.csv")

[set ProjectsOwnParameterFile? false] ; to make sure that under default setting, the default ;(System) parameter file is used

if ProjectsOwnParameterFile? = true

_]

set MyParametersFile (word remove " SYSTEM " MyMap) ; the parameter file of this project has the same name as the map used, but make sure, system files cannot be overwritten

set MyParametersFile (word remove ".png" MyParametersFile) ; MyMap is an image file but MyParametersFile is not - delete extension!

set MyParametersFile (word MyParametersFile " Parameters.csv") ; identifier for parameter file and correct extension

_]

PanelSettingProc

ReadAllParametersProc

ParametersProc

CreateFoodsourcesProc

CreateSpeciesProc

CreateBadgersProc

CreateInitialQueensProc

UpdateMorning_Proc

CreateSignsProc

OutputDailyProc

if ShowGrid? = true

[

ask patches with [remainder pxcor round (Gridsize * Scaling_NLpatches/m) = 0] [set pcolor black

]

ask patches with [remainder pycor round (Gridsize * Scaling_NLpatches/m) = 0] [set pcolor black

]

ask patch 290 5 [set plabel-color black set plabel word Gridsize " m"]

]

if (MyMap = " SYSTEM Example Farm.png" and MyParametersFile = " SYSTEM Example Farm Parameters.csv")

[set ProjectsOwnParameterFile? true] ; to make sure that under default setting, the default ;(System) parameter file is used

if count foodsources with [patchtype = "undefined"] > 0 [AssertionProc "Undefined habitat 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.

end

, *****
,

to ReadAllParametersProc

```
:: if ProjectsOwnParameterFile? = true:
:: MyParametersFile is set to the project's default parameter file name
("projectname Parameters.csv").
:: If this file exists, it is loaded, if it does not exist, it is created from " SYSTEM Parameters.csv"
::
:: if ProjectsOwnParameterFile? = false:
:: A) if MyParametersFile exists, it is loaded as parameter file
:: B) otherwise, MyParametersFile is set to " SYSTEM Parameters.csv" (and loaded, without being
saved under a new name)
::

let newParameterfileNeedsToBeCreated? false
let parametersDataCSV [] ; csv:from-file MyParametersFile ; a csv input file is loaded and saved
ifelse file-exists? MyParametersFile
[ set parametersDataCSV csv:from-file MyParametersFile ]
[
ifelse file-exists? " SYSTEM Parameters.csv"
[
output-print "MyParametersFile was created from ' SYSTEM Parameters.csv'"
set parametersDataCSV csv:from-file " SYSTEM Parameters.csv"
set MyParametersFile " SYSTEM Parameters.csv"
]
]

if ProjectsOwnParameterFile? = true [ set newParameterfileNeedsToBeCreated? true ]
[
[
output-print "Neither the specified MyParametersFile nor ' SYSTEM Parameters.csv' do exist in
this folder!"
user-message "Can't find input file ' SYSTEM Parameters.csv'"
]
]
let header item 0 parametersDataCSV ; first line of the input file is the header
set AllParametersList header
let parameterDataList item 1 parametersDataCSV

foreach header
[
let nextCommand ""
let newEntry (word item (position (word ?) header) parameterDataList)
if member? "\"\"\" newEntry and ? != "MyFarmlandPatches" ; if margins are added doubled
double-quotes can occur. They need to be replaced by single double-quotes. Exception for
MyFarmlandPatches, which actually can be "" CHANGE THIS!! set MyFarmland to "none"
```

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 newEntry remove "\" newEntry  
  set newEntry (word "\" newEntry "\" )  
]  
set nextCommand (word "set " ? " " newEntry)  
run nextCommand  
]  
if newParameterfileNeedsToBeCreated? = true [ SaveLoadSettingsProc "Save!" "" ]  
end
```

```
. *****  
.
```

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

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 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

```

```

, *****
;

```

to Go

```

if AssertionViolated = true

```

```

[

```

```

ask patches [ set pcolor red ]

```

```

user-message "Assertion violated!"

```

```

stop

```

```

]

```

```

; 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

```

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 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
 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 ThresholdLevelREP [thType situation]

;TYPES: eggLaying pollenForaging nectarForaging nursing

;SITUATIONS: egg worker youngQueen QueenInitiationPhase QueenSocialPhase Psith

let th -1

;Egg Laying

if thType = "eggLaying"

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 = "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]
  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

|
, *****
;
```

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 CreateInitialQueensProc

```
let newQueens 0
let counter 0

foreach SpeciesList ; lists bee species present
—BeeSpeciesInitialQueensList ; e.g. [ ["B hortorum" 100] ]"
—if ? = "B_lapidarius" [ set newQueens B_lapidarius ] ; numbers specified on interface
—if ? = "B_pascuorum" [ set newQueens B_pascuorum ]
```

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 ? = "B_terrestris" [set newQueens B_terrestris]500]]
—if ? = "B_hortorum" [set newQueens B_hortorum]
—if ? = "B_hypnorum" [set newQueens B_hypnorum]
—if ? = "B_pratorum" [set newQueens B_pratorum]
—if ? = "Psithyrus" [set newQueens N_Psithyrus]

```

```

[
let modelledSpecies ?[item 0 ?

```

```

if InitialQueensModifierBehaviorSpace = 0 [ set InitialQueensModifierBehaviorSpace 1 ]
set newQueens round (item 1 ? * InitialQueensModifierBehaviorSpace)

```

```

create-bees newQueens ; newQueens = number of new queens created here
[
set shape "circle"
if ShowQueens? = false [ hide-turtle ]
set size QueenSymbolSize
set adultAge 180 ; queens have hibernated (exact age doesn't matter)
set broodAge 36 ; (exact age doesn't matter)
set color red
set brood? false
set caste "queen"
set mated? true
set number 1
set ploidy 2
set mtDNA random-float 139.9 ; i.e. within the range of Netlogo colours
set allelesList list (random-float 139.9) (random-float 139.9)
set spermathecaList [] ;list (allele)
set spermathecaList fput (random-float 139.9) spermathecaList
set colonyID -1 ; i.e. does not belong to any colony yet
let speciesIDmemo -1
let speciesNameMemo "noName"
ask one-of Species with [ name = modelledSpecies ]
[
set speciesIDmemo who
set speciesNameMemo name
]
set speciesID speciesIDmemo
set speciesName speciesNameMemo
set stage "adult"
set thEgglaying ThresholdLevelREP "eggLaying" "QueenInitiationPhase"
set thForagingNectar ThresholdLevelREP "nectarForaging" "QueenInitiationPhase"
set thForagingPollen ThresholdLevelREP "pollenForaging" "QueenInitiationPhase"
set thNursing ThresholdLevelREP "nursing" "QueenInitiationPhase"
set activity "hibernate"
set activityList [ ]
set personalTime_s random (2 * 3600) + (GetUpTime_s - 3600) ; = Start_time_s +- 1hr (i.e.
between 7:00 and 9:00 am)
let yearEndSeason [seasonStop] of OneSpecies speciesID ; prevent bees from setting
emergingDate past the end of season
while [ emergingDate <= 0 OR emergingDate >= yearEndSeason]

```


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 emergingDate round random-normal [ emergingDay_mean ]
of OneSpecies speciesID [emergingDay_sd] of OneSpecies speciesID ]
; emerging from hibernation next year on day "emergingDay_mean" (+- s.d.)

set currentFoodsource -1
set nectarsourceToGoTo -1
set pollensourceToGoTo -1
set pollenForager? false
set knownMasterpatchesNectarList [ ]
set knownMasterpatchesPollenList [ ]

; determination of the queen's weight:
let minWeight_mg [ devWeight_Q_PupationMin_mg ] of oneSpecies speciesID
let maxWeight_mg [ devWeight_Q_PupationMax_mg ] of oneSpecies speciesID
let meanWeight_mg (maxWeight_mg + minWeight_mg) / 2
let sd_weight (maxWeight_mg - minWeight_mg) / 4 ; mean +- 2xSD > 95%

set weight_mg random-normal meanWeight_mg sd_weight
if weight_mg > maxWeight_mg [ set weight_mg maxWeight_mg ]
if weight_mg < minWeight_mg [ set weight_mg minWeight_mg ]
set glossaLength_mm ProboscisLengthREP
set cropvolume_myl CropAndPelletSizeREP "nectar"
set pollenPellets_g CropAndPelletSizeREP "pollen"
]
]
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
```

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 result = 0 [ AssertionProc "Assertion violated: Error in CropAndPelletSizeREP" ]
report result ; units: ul for nectar, g for pollen!
end
```

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

to CreateBadgersProc

; suggestions for initial number of badgers: zero, intermediate (>0–3 setts km2) and high (>3 setts km2),

; Reilly & Courtenay 2007 (Preventive Veterinary Medicine 80 (2007) 129–142)

let burrowHabitatsList ["Scrub"] ; habitats badgers can nest in

let distanceLimit_m 300 ; badgers cannot be created within this distance of a current sett
; (Kruuk 1978, J. Zool., Lond.184, 1-19; Fig. 2)

let memoX 0

let memoY 0

; convert distance to netlogo patches:

let distance_patches distanceLimit_m * SCALING_NLpatches/m

; agentset of suitable foodsources:

let burrowFsSet FoodSources with [(member? patchtype burrowHabitatsList) AND masterPatch?]

create-badgers N_Badgers ; create the badgers

[

ifelse count burrowFsSet > 0 ; check for suitable foodsource

[

let chosenFs one-of burrowFsSet ; one of the suitable patches is chosen..

ask chosenFs [set memoX pxcor set memoY pycor] ; its location is saved..

setxy memoX memoY ; .. the badger moves to the location of the patch

set size 9 * MasterSizeFactor

set shape "badger"

; recreate the agentset, only taking masterpatches without any badgers in a certain radius into

account:

set burrowFsSet FoodSources with [(member? patchtype burrowHabitatsList) AND

masterPatch? AND

count badgers-here = 0 AND

count badgers in-radius distance_patches = 0

]

]

[DieProc "Badger: not enough habitat!"]; no badgers if there is no habitat for their burrows!

]

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

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 encounterProb 0.19 ; probability 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

```

; *****

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
    ]
  ]
]

```

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

```

```

, *****
,

```

to AssertionProc [message]

```

  show message
  set AssertionMessage message
  set AssertionViolated true
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 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

; *****
```

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
```

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 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 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 DevelopmentProc

```
ask bees
[
```

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 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
]
[
  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 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" ]

```

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 Development_PupaeProc

; procedure checks if pupae develop into adults

set cumTimePupa_d cumTimePupa_d + 1 ; potential output (time spent as pupa)

if caste = "worker"

[

; Development factors pupae: age & incubation:

if cumIncubationReceived_kJ >= [devIncubationEmergingTH_kJ] of OneSpecies speciesID

and broodAge >= [devAgeEmergingMin_d] of OneSpecies speciesID

[

set stage "adult"

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 cumIncubationReceived_kJ >= [devIncubationEmergingTH_kJ] of OneSpecies speciesID

and broodAge >= [devAgeEmergingMin_d] of OneSpecies speciesID

[

set brood? false

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 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
    [
        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
    ]
]

```

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

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

to Development_LarvaeProc [whoCol]

```
; procedure checks if larvae develop into pupa
; Development factors larva: age, incubation and weight - option to develop into queen!
set cumuTimeLarva_d cumuTimeLarva_d + 1
if caste = "undefined" and cumuTimeLarva_d = [ dev_larvalAge_QueenDetermination_d ] of
OneSpecies speciesID ; age of determination is independent of 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 cumuIncubationReceived_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]
]
```

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 = "queen" ;
  and cumullIncubationReceived_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"
  set color red
  let memoNumber number
  ask colony colonyID [set totalPupaeProduced totalPupaeProduced + memoNumber]
]

```

end

, *****

to Development_EggsProc

```

; ; procedure checks if eggs develop into larvae
set cumulTimeEgg_d cumulTimeEgg_d + 1
if cumullIncubationReceived_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 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
  [

```

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 eggDeathsIncubation eggDeathsIncubation + memoNumber]
DieProc "Egg: broodAge > devAgeHatchingMax_d"
]
if stage = "larva"
[
  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)
[

```

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 stage = "egg"
    [
        ask colony colonyID [set broodDeathsCP broodDeathsCP + memoNumber]
        DieProc "Egg: CP!"
    ]
]
]
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 ]
    set breed deadCols
    set size 1
    set label ""
    set color white
    set shape "circle"

```

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 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 CheckNumbersProc

```

let totalInitalBees 0
B_lapidarius
+ B_pascuorum
+ foreach BeeSpeciesInitialQueensList ; e.g. [ "B_hortorum" 100 ] [ "B_terrestris" 500 ]
+ B_hortorum
+ B_hypnorum
+ B_pratorum
+ N_Psithyrus

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

let totalBeesPresent sum [ number ] of bees

if totalInitalBees + 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" ]

```

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*.

```

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

```

```

.... *****
,,,,

```

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-STEWART: a research and decision support software for effective land management to promote bumblebee populations. Methods in Ecology and Evolution.

to-report StimNursingREP

```
; asked by colony
; 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 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
```


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 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 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

```

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*.

```

        EggLayingProc
    ]
    if activity = "nursing" [ set activityList lput "NURSE" activityList
        BroodIncubationProc
    ]
    if activity = "pollenForaging"
        [ set activityList lput "P-FOR" activityList
        ForagingProc
        ]
    if activity = "nectarForaging"
        [ set activityList lput "N-FOR" activityList
        ForagingProc
        ]
end

, *****

```

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)

```

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*.

```

]
[ ; 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!

```

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

```

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 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 cumulTimeEgg_d 0 ; Set to 0, not mother's value
set cumulTimeLarva_d 0 ; Set to 0, not mother's value
set cumulTimePupa_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
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 cumulIncubationReceived_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
```

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 here⁴)

; 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 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

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*.

```

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
[
  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

```

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.

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

```

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 nectar_myl nectarMax_myl
    set pollen_g pollenMax_g
  ]
  [
    set nectar_myl 0
    set pollen_g 0
  ]
  set NectarAvailableTotalToday\_l NectarAvailableTotalToday\_l + (nectar_myl / (1000 * 1000))
  set PollenAvailableTotalToday\_kg PollenAvailableTotalToday\_kg + (pollen_g / 1000)
]
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-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
```


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 compDate round (eusocialPhaseDate + y)
if compDate - eusocialPhaseDate > latestCPafter_d
  [ set compDate eusocialPhaseDate + latestCPafter_d ]
report compDate
end

```

```

, *****
,

```

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" ]

```

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 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 ]
[ 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 number of adult males..
  let queensHere totalQueensProduced ; .. and adult queens ever produced by this dying colony..
  ask patch-here
  [
```

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 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
    ]
    [
        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-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)" ]

```

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 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
```

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

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."
```

```
;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"]
[
```

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 != "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
```

```
        [ 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 DrawCohortsProc

```
ask 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*.

```
[
  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
```

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

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
```

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 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
```

```
]
```

```
    ; 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
```

```
]
```

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*.

```
; 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
```

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

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 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
]
[
```


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 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-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

```

```

, *****
,

```

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 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!
      [ 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
      ]
    ]
  ]
]
```

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

```

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 cumulPollenVisits cumulPollenVisits + memoNumber ; all pollen visits at patch, ever
    ]
]
[
    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
                                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-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

```

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 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

```

```

, *****
;

```

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 ]

```

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: if pollen forager
if handlingTime_s > maxHandlingTime_s [ set handlingTime_s maxHandlingTime_s ]
report handlingTime_s
end
```

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

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
```

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*.

```
[
; 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
)
    * [ 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_PatchChoiceProc [currentTripDuration_s]

; determines if bees are still happy with their current food source (based on the duration of the trip).
 If not, they will either switch to the best layer/flowerspecies at the
 ; current patch or - if they already forage at the best layer/flowerspecies, they will switch to another
 patch/'layergroup' they know or search for a new one
 ; (note: expectation_Nectar/PollenTrip_s is 0 for a novice forager, hence they are likely to search
 new patches until they get more experienced)
 ; Ref: Wiegmann et al 2003, *Physiology & Behavior* 79 (2003) 561– 566

let preferenceClosePatchesProb 0.9 ; heuristically determined to result in highest numbers of
 hibernating queens
 let happy? true ; defines whether or not a bee is still happy with her current food source

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 gotoNewLayergroup? false ; whether or not the bee is going to exploit a different flower patch
let bestLayer -1 ; ID (who) of best food source in the current flower patch. As long as the bee is
happy, it doesn't matter which foodsource is the best layer
let myExpectation_s expectation_NectarTrip_s ; expectation of a bee about the duration of the trip
let myKnownMasterpatchesList knownMasterpatchesNectarList
let searchProbBase 0.1
```

```
if pollenforager? = true ; some changes if a bee is a pollen forager and not a nectar forager
[
  set myExpectation_s expectation_PollenTrip_s
  set myKnownMasterpatchesList knownMasterpatchesPollenList
]
let myMasterpatchID -1
let myCurrentPatchPosition -1
```

; the longer a trip takes in comparison to a bees' expectation, the higher is the probability to become unhappy. Expectations are then recalculated as the mean of the duration of current trip and the previous expectation

ifelse activity = "returningEmpty"

```
[ set gotoNewLayergroup? true ] ; unsuccessful bees always search for a new flower
patch/'layergroup'
```

```
[
  set myMasterpatchID [ masterpatchID ] of foodsource currentFoodsource
  if position myMasterpatchID myKnownMasterpatchesList = false [ AssertionProc ("No number for
myMasterpatchID (Foraging_PatchChoiceProc) Possible reason: 2 flower patches (masterpatches,
not just layers) at same location?") ]
  ; the "position" (in the list) of the currently used 'layergroup' in the myKnownMasterpatchesList:
  set myCurrentPatchPosition position myMasterpatchID myKnownMasterpatchesList ; NeLogo
command "position": "On a list, reports the first position of item in list, or false if it does not
appear."
```

```
if myCurrentPatchPosition = false [ set myCurrentPatchPosition -1 ] ; to avoid an error if
myMasterpatchID is not part of the myKnownMasterpatchesList
```

```
; determine whether or not a bee becomes unhappy with her current foodsource:
let unhappyProb (currentTripDuration_s - myExpectation_s) / currentTripDuration_s
; bee compares her expectations on trip duration with the actual duration
if unhappyProb < 0 [ set unhappyProb 0 ] ; no negative probability
; the probability to search a new patch then depends on searchProbBase
; and her degree of unhappiness:
if random-float 1 < (searchProbBase + unhappyProb) [ set happy? false ]
let newExpectation (myExpectation_s + currentTripDuration_s) / 2 ; new expectation take
duration of current and previous trips into account
; new expectation take duration of current and previous trips into account
ifelse pollenforager? = true
[ set expectation_PollenTrip_s newExpectation ] ; as expectations for pollen trips
[ set expectation_NectarTrip_s newExpectation ] ; .. or for nectar trips
]
```

```
if happy? = false
```

```
[
```

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 activityList lput ":" activityList ; unhappy smiley added to activityList
ifelse pollenforager? = true ; splitting returningUnhappy into two types
[ set activity "returningUnhappyP" ]
[ set activity "returningUnhappyN" ]
set bestLayer Foraging_bestLayerREP currentFoodsource
ifelse bestLayer = currentFoodsource
[ set gotoNewLayergroup? true ] ; if bee is already foraging from the best layer, it will search for
a completely new patch/'layergroup'
[ set currentFoodsource bestLayer ] ; .. otherwise it will stay at the current patch but switch to
the best foodsource/flowerspecies here
]

if gotoNewLayergroup? = true
[
set activityList lput "..." activityList
let newPatchPosition -999
ifelse random-float 1 < preferenceClosePatchesProb ; the new patch has a similar distance to the
colony as the old patch, but patches closer to the colony are preferred
[ set newPatchPosition random (myCurrentPatchPosition + 1) - 1 ] ; a random position < current
position, includ. 0 and -1 (-1 results in search of a new 'layergroup'. Prob. decreases, the more
patches are known)
[ set newPatchPosition myCurrentPatchPosition + 1 ]
ifelse newPatchPosition < 0 or newPatchPosition >= length myKnownMasterpatchesList
[
set currentFoodsource -1 ; bees will search for a completely new foodsource/'layergroup'
]
[ set currentFoodsource Foraging_bestLayerREP item newPatchPosition
myKnownMasterpatchesList ] ; bee goes to a patch ('layergroup') it already knows and chooses the
best foodsource (layer) there
]
ifelse pollenforager? = true
[ set pollensourceToGoTo currentFoodsource ] ; the (new) current foodsource will be used for the
next pollen foraging trip
[ set nectarsourceToGoTo currentFoodsource ] ; the (new) current foodsource will be used for the
next nectar foraging trip
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
```

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 tripDuration_s 0
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 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

```

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 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

```

, *****

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

```

, *****

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 ]

```

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 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 " " ]
  ] ; "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
```

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 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
]
```

```
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 spermathecaList
      [ 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 ;
]
```

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 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
  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
```


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.

```
- 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  
    — 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"
```

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-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
]
```

```
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 > 0caste = "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.

```
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
    —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" ]
```

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*.

```
—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" ]  
—  
  
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" ]
```

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.

```
—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"]  
]
```

```
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
```

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 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) ]
]
]
]

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"
```

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-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
]
]

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
    ]
]
```

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 = "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"

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

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-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 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 ]
[
  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 Foraging_periodREP

```
let foragingPeriod_s -1
let foragingHoursList [ ]
```

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 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
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 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 CreateSignsProc

```
create-signs 1 ; Weather symbol: Sun
[
  setxy max-pxcor - 6 max-pycor - 16
  set shape "sun"
  set size 11
  set color 44.2
  hide-turtle
```

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.

]

create-Signs 1 ; Weather symbol: Cloud

[

setxy max-pxcor - 10 max-pycor - 17

set shape "cloud"

set size 11

set color grey - 2

hide-turtle

]

create-Signs 1 ; Symbol for FIND-Button

[

set color red

set shape "circletarget"

set size 30

hide-turtle

]

[create-brushSigns 1 ; Symbol for brush sign](#)

[

[ifelse CircularBrush? = true](#)

[\[set shape "circle" \]](#)

[\[set shape "square" \]](#)

[hide-turtle](#)

]

end

;

;===== BUTTONS === BUTTONS === BUTTONS === BUTTONS === BUTTONS ===

;

[to ActivityListButton](#)

[:: to ActivityListButtonProc](#)

[::type "day: " type day print " \(species caste colony bee age activities\) "](#)

[::](#)

[:: foragers sorted by "colonyID" and then by "who"](#)

[::foreach sort-on \[speciesID * 1000000000000000 + colonyID * 10000000 + who \] bees with \[stage = "adult" and colonyID >= 0\]](#)

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 ?
—{
:: [
::   type speciesName type " "
::   if caste = "worker" [ type "W" type " " ]
::   if caste = "queen" and mated? = true [ type "Q" type " " ]
::   if caste = "queen" and mated? = false [ type "q" type " " ]
::   if caste = "male" and mated? = false [ type "M" type " " ]
::   type colonyID type " "
::   type who type " "
::   type adultAge type " "
::   print activityList ] ]
::end
```

. *****

```
to-DefaultProc
-;set RAND_SEED 1
-set AbundanceBoost 1
-set B_hortorum 0
-set B_hypnorum 0
-set B_lapidarius 0
-set B_pascuorum 0
-set B_pratorum 0
-set B_terrestris 500
-set Backgroundcolour 5
-set ChooseInputFile "BBH-T_Suss1.txt"
-set ChooseInputMap "BBH-I_Suss1.png"
-set COLONIES_IBM 0
-set FlowerspeciesFile "BBH-Flowerspecies_Suss.csv"
-set FoodSourceLimit 25
-set ForagingMortalityFactor 1
-set ForagingMortalityModel "high"
-set GenericPlot1 "Species total adult queens"
-set GenericPlot2 "Species N colonies"
-set GenericPlot3 "Food available"
-set GenericPlot4 "Colony structures"
-set GenericPlot5 "Species total adults"
-set Gridsize 500
-set INPUT_FILE "BBH-T_Suss1.txt"
-set InputMap "BBH-I_Suss1.png"
-set InspectTurtle 1
-set KeepDeadColonies? true
-set Lambda_detectProb 0.005
-set MapAreaIncluded "complete"
-set MasterSizeFactor 1
-set MaxHibernatingQueens 10000 ; reduces number of hibernating queens (if exceeding) on 1st
January to this number
-set MergeHedges? true
-set MinSizeFoodSources? TRUE
```

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 N_Badgers 0
set N_Psithyrus 0
set RemoveEmptyFoodSources? TRUE
set SexLocus? false ; true
set ShowCohorts? true
set ShowDeadCols? false
set ShowFoodsources? true
set ShowGrid? false
set ShowInspectedColony? false
set ShowMasterpatchesOnly? false
set ShowNests? true
set ShowPlots? true
set ShowQueens?
, *****
/

true
=set ShowSearchingQueens? true
set ShowWeather? true
set SpeciesFilename "BBH BumbleSpecies_UK_01.csv"
set StopExtinct? true
set UnlimitedMales? true
set Weather "Constant 8 hrs"
set WinterMortality? true
set X_Days 90
end

, *****
/

```

to VersionTestProc

; to test whether the model was changed

let expectedValue 28690 8560; (46703: 2020-06-03) ; (29447: 2020-01-28) ; 12485 ; 12485: 2018-07-27

~~DefaultProc~~

set GenericPlot1 "Number of colonies for different species" ; not really needs setting, as output runs with "local randomness"

set MyMap " SYSTEM Sussex1.png"

set MyParametersFile " SYSTEM Parameters.csv"

set ProjectsOwnParameterFile? false

ClearProc

set RAND_SEED 1

~~set random-seed RAND_SEED~~

if ProjectsOwnParameterFile? = true

└

set MyParametersFile (word remove " SYSTEM " MyMap) ; the parameter file of this project has the same name as the map used, but make sure, system files cannot be overwritten

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 MyParametersFile (word remove ".png" MyParametersFile) ; MyMap is an image file but  
MyParametersFile is not - delete extension!  
set MyParametersFile (word MyParametersFile " Parameters.csv") ; identifier for parameter file  
and correct extension  
]   
PanelSettingProc  
ReadAllParametersProc  
set BeeSpeciesInitialQueensList [ ["B_hortorum" 20  
-set-] ["B_hypnorum" 20  
-set-] ["B_lapidarius" 20  
-set-] ["B_pascuorum" 20  
-set-] ["B_pratorum" 20  
-set-] ["B_terrestris" 100  
-set N-] ["Psithyrus" 20]]  
set N_Badgers 5  
  
Setup  
ParametersProc  
CreateFoodsourcesProc  
CreateSpeciesProc  
CreateBadgersProc  
CreateInitialQueensProc  
UpdateMorning_Proc  
CreateSignsProc  
OutputDailyProc  
if ShowGrid? = true  
]   
ask patches with [ remainder pxcor round (Gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor black  
]   
ask patches with [ remainder pycor round (Gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor black  
]   
ask patch 290 5 [ set plabel-color black set plabel word Gridsize " m" ]  
]   
repeat 2 * 365  
[  
  Go  
  if AssertionViolated = true  
  [  
    ask patches [ set pcolor red ]  
    stop  
  ]  
]  
- let testValue TotalBeesEverProduced + TotalHibernatingQueens + TotalMales + TotalAdultWorkers  
+ TotalFoodSources  
type testValue type " — Difference: " print testValue — expectedValue  
  
ifelse testValue = expectedValue  
- [ user-message (word "No deviation detected from the publishedofficial Beesteward (2020) version  
of Bumble-BEEHAVE (2017)-(Test value: " expectedValue ")") ) ]  
-[
```

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.

```
type testValue type " Difference: " print testValue - expectedValue  
_user-message "CHANGES MADE TO THE MODEL OR INPUT FILES!"  
_  
end  
  
; *****  
  
;;; BEESCOUT - BEESCOUT - BEESCOUT - BEESCOUT - BEESCOUT - BEESCOUT - BEESCOUT -
```

to BS_ColourCorrectionsMapProc

```
ask patches  
[  
  set originalColor pcolor ;; the color of the (grid) patch in the original file  
  set plabel-color black  
  set flowerPatchID -1  
  if remainder pcolor 10 < Black_th [ set pcolor black ]  
  if remainder pcolor 10 > White_th [ set pcolor white ]  
]  
  
ask habitats  
[  
  let memoColorID habitatColourID  
  let memoColorMin colourRangeMin  
  let memoColorMax colourRangeMax  
  
  ask patches with [ pcolor >= memoColorMin and pcolor <= memoColorMax and (pcolor !=  
BorderColor)]  
  [ set pcolor memoColorID ]  
]  
ask patches [ set patchColor pcolor ]  
end  
  
; *****
```

to ButtonLoadExistingMapProc

```
; only called by button!  
_user-message "Choose one of the maps you have created (image '.png' file)"  
_let myChosenMap user-file ; reports the complete path - however, only the filename is required!  
_if myChosenMap != false ; 2019-10-14 (*SOLITARY*)  
_  
_set myChosenMap reverse myChosenMap ;; as position reports only the first (and not the last)  
occurrence, string is reversed, e.g. gnp.mraF_elpmaxE\DRAWETS_EEB\C
```

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 cutHere position "\\\" myChosenMap ; this is the filename + extension, but reversed  
set myChosenMap substring myChosenMap 4 cutHere ; reversed filename without extension, ;  
e.g. mraF elpmaxE  
set myChosenMap reverse myChosenMap ; this is the actual filename  
set myChosenMap remove " Foodsources" myChosenMap ; in case a foodsource (txt) file was  
chosen  
set myChosenMap remove " Parameters" myChosenMap ; in case a parameter (csv) file was  
chosen  
set MyParametersFile (word myChosenMap " Parameters.csv")  
set MyMap (word myChosenMap ".png")
```

Setup

```
]   
end
```

```
. *****
```

```
::to ButtonDefineFarmareaProc ; Currently not in use  
:: ask foodsources  
:: [  
:: ifelse pcolor = black  
:: [  
:: set farmland? true  
:: set MyFarmlandPatches (word MyFarmlandPatches " " who " ")  
:: ]  
:: [ set farmland? false ]  
:: ]  
:: ask patches with [ pcolor = black ]  
:: [  
:: set pcolor pcolorSave  
:: ]  
::end
```

```
. *****
```

to ButtonUseCurrentMapProc

```
let myName (word remove ".png" MyMap)  
let prelimName user-input "Set a new name for your map (or leave blank to overwrite): "  
if prelimName != "" [ set myName prelimName ]  
set myName (word remove " SYSTEM " myName) ; make sure, system files cannot be  
overwritten!  
set MyMap (word myName ".png")  
ask turtles [ hide-turtle ] ; we don't want to see bee etc. on the map!  
export-view (word myName ".png") ; image file of the map is created  
SaveLoadSettingsProc "Save!" "" ; settings are saved  
BS_ImportNewMapProc FALSE ; FALSE: no user input required  
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 BS ImportNewMapProc [userInput?]

```
let readFromFile? false  
clear-all  
stop-inspecting-dead-agents  
reset-ticks  
if RAND SEED != 0 [ random-seed RAND SEED ]  
  
ReadAllParametersProc  
  
if user-yes-or-no? "Select a new habitat input file for this map? (If uncertain, press 'No')"  
[ set HabitatsFile FilenameREP true ] ; true: filename has extension already added  
  
CreateHabitatsProc  
  
set BorderColor 125.12345678987654 ; a unique color most likely not occurring on the map  
(~magenta)  
set OutputWordResult ""  
ask patches [ set pcolor BorderColor ] ; pcolor set to a unique color, to determine the borders of the  
map, after the map has been loaded  
set PatchColoursList []  
  
foreach sort habitats  
[  
ask ? [ if habitatSwitchedOn? [ set PatchColoursList lput habitatColourID PatchColoursList ] ]  
]  
ifelse userInput? = true  
[  
ifelse user-yes-or-no? "Load scan/image file? (click 'No' to load a text file)"  
[  
set MyMap FilenameREP true  
import-pcolors MyMap  
]  
[  
set readFromFile? true  
BS_ReadMapFromTextProc ; import-pcolors takes place in this procedure  
]  
[ import-pcolors MyMap ] ; (i.e. if userInput? = false)  
  
if readFromFile? = false [ BS_Dialogue_ScalingProc ]  
BS_ColourCorrectionsMapProc  
BS_AnalyseProc  
  
ask patches [set pcolor grey ]  
ask patches with [ member? patchColor PatchColoursList = true] [ set pcolor patchColor ]  
SaveLoadSettingsProc "Save!" "" ; create new Parameter file
```

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.

[BS_WriteBumbleBeehaveOutProc userInput?](#)
[Setup](#)

[end](#)

[. *****](#)

[to-report DataFromTextMapREP](#)

[; used when map is read in from TextMap with the new, GIS compatible format, result is the relevant number in the data line](#)

[let dataline file-read-line](#)

[let datalineRev reverse dataline](#)

[let firstBlankPosition position " " datalineRev](#)

[let resultRev substring datalineRev 0 firstBlankPosition ; incl. position 0, excludes position 3](#)

[set datalineRev remove " " resultRev](#)

[let result read-from-string reverse datalineRev](#)

[report result](#)

[end](#)

[. *****](#)

[to BS ReadMapFromTextProc](#)

[let textFileToOpen FilenameREP false](#)

[set MyMap \(word textFileToOpen ".png"\)](#)

[file-open \(word textFileToOpen ".txt"\)](#)

[set GIS_ncols DataFromTextMapREP](#)

[set GIS_nrows DataFromTextMapREP](#)

[set GIS_xllcorner DataFromTextMapREP](#)

[set GIS_yllcorner DataFromTextMapREP](#)

[set BS_Scaling_m/NLpatches DataFromTextMapREP](#)

[set GIS_NoDataValue DataFromTextMapREP](#)

[let leftMargin \(max-pxcor - GIS_ncols\) / 2](#)

[let bottomMargin \(max-pycor - GIS_nrows\) / 2](#)

[foreach sort patches](#)

[\[](#)

[ask ?](#)

[\[](#)

[if \(pxcor >= leftMargin and pxcor < GIS_ncols + leftMargin\) and \(pycor >= bottomMargin and pycor < GIS_nrows + bottomMargin\) and file-at-end? = false](#)

[\[](#)

[set pcolor file-read](#)

[\]](#)

[\]](#)

[\]](#)

[file-close](#)

[ask patches](#)

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 originalColor pcolor ;; the color of the (grid) patch in the original file
  set plabel-color black
  set flowerPatchID -1
  set patchColor pcolor ;; saves the color after rounding
]
end
```

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

to BS Dialogue ScalingProc

```
set BS_Scaling_m/NLpatches 1 / Scaling_NLpatches/m ; scaling in BEESCOUT is 1 / scaling in
Bumble-BEEHAVE
let distance_NLpatches 0
let p1 nobody
let p2 nobody
let memoColorP1 0
let memoColorP2 0

let userChoice1 user-one-of "Choose a method to define the scaling of your map"
[
  "Scaling is already defined - no changes needed"
  "Direct Input"
  "Distance left - right side"
  "Distance top - bottom"
  "Distance any two points"
]

if userChoice1 = "Distance left - right side" [ set distance_NLpatches max-pxcor - min-pxcor ]
if userChoice1 = "Distance top - bottom" [ set distance_NLpatches max-pycor - min-pycor ]
if userChoice1 = "Distance any two points"
[
  let userHappy? false
  let pointColor red
  user-message "Please select now two reference points of known distance via two mouse clicks
(make sure speed-slider is set to 'normal speed')"
```

```
  while [ userHappy? = false ]
  [
    while [ p1 = nobody ]
    [
      if mouse-down?
      [
        set p1 patch mouse-xcor mouse-ycor
        ask p1
      ]
      set memoColorP1 pcolor
    ]
  ]
]
```

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 pcolor pointColor
  ]
]
display
]
wait 0.5
while [ p2 = nobody ]
  [
    if mouse-down?
    [
      set p2 patch mouse-xcor mouse-ycor
      ask p2 ; patch mouse-xcor mouse-ycor
    ]
    set memoColorP2 pcolor
    set pcolor pointColor
  ]
]
display
]

let userChoiceHappiness user-one-of "Are you happy with your reference points?"
[
  "Yes - continue"
  "No - do it again"
  "No - do it again with different point colors"
]

if userChoiceHappiness = "Yes - continue" [ set userHappy? true ]
if userChoiceHappiness = "No - do it again" or userChoiceHappiness = "No - do it again with
different point colors"
[

  ask p1 [ set pcolor memoColorP1 ]
  ask p2 [ set pcolor memoColorP2 ]
  set p1 nobody
  set p2 nobody
  set memoColorP1 0
  set memoColorP2 0
  display
]
if userChoiceHappiness = "No - do it again with different point colors"
[
  set pointColor read-from-string user-one-of "Choose the colour of your reference points:"
  [
    "Black"
    "White"
    "Grey"
    "Blue"
    "Green"
    "Yellow"
  ]
]

```

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.

```

  "Red"
  "Green"
]
]
] ; while user unhappy
ask p1 [ set distance NLpatches distance p2 ]
] ; "Distance any two points"

if userChoice1 != "Scaling is already defined - no changes needed" and userChoice1 != "Direct Input"
[ set BS_ScaleDistance_m read-from-string user-input "Real distance [m] is (e.g. 150): " ]

if userChoice1 = "Distance any two points"
[
ask p1 [ set pcolor memoColorP1 ]
ask p2 [ set pcolor memoColorP2 ]
]
if userChoice1 != "Scaling is already defined - no changes needed" and userChoice1 != "Direct Input"
[ set BS_Scaling_m/NLpatches BS_ScaleDistance_m / distance NLpatches ] ;; real distance [m]
divided by distance of grid points

if userChoice1 = "Direct Input"
[ set BS_Scaling_m/NLpatches read-from-string user-input "Scaling is [m/grid cell] (e.g. 25): " ]
end

. *****
/
```

to BS_AnalyseProc

```

ask turtles with [ breed != habitats ] [ die ]
ask turtles [ hide-turtle ]

let currentColor 0
let currentPatchID -1
let flowerPatchCounter 0
set Repetitions round (MaxPatchRadius_m / BS_Scaling_m/NLpatches) ; # of repetition depends
on the scale of the landscape

foreach sort patches
[
ask ? ;; determines flowerpatches: same-coloured (nlogo-)patches with ID -1 are searched, it gets
a new ID and all connected (nlogo-)patches with the same colour get the same ID
[
if member? patchColor PatchColoursList
[
if flowerPatchID < 0 ;; if patch is not identified yet (flowerPatchID is patches-own variable, set to
-1 in BS_ColourCorrectionsMapProc)
[
set flowerPatchID flowerPatchCounter
```

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 currentColor patchcolor ;; colour of the flower patch is the colour of the
"firstPatchOfFlowerpatch"
set firstPatchOfFlowerpatch true
set currentPatchID flowerPatchID

let search-patches patches in-radius Repetitions
repeat Repetitions
[
  ask search-patches with [(patchcolor = currentColor) and (flowerPatchID = currentPatchID)] ;;
  connected (nlogo-)patches around the firstPatchOfFlowerpatch are searched defined as part of this
  flower patch
  [
    ask neighbors with [(patchcolor = currentColor) and (flowerPatchID = -1)]
    [ set flowerPatchID currentPatchID ]
  ]
]
set flowerPatchCounter flowerPatchCounter + 1
]
]
set pcolor lime ; to show progress of map analysis
]
display
]

ask patches with [ flowerPatchID >= 0 ] [ set flowerPatchID flowerPatchID + count turtles ] ; in the
original BEESCOUT model, all turtles were cleared at the beginning of
; this procedure, now, habitat-turtles are not cleared, hence flowerPatchID needs to be increased
by the number of (habitat) turtles

set Npatches currentPatchID + 1 ; as 1st patch has id 0
BS CreatePatchStatisticsProc ;; creates "patchStatistics" (turtles) to store data of the flower
patches ; CALL A PROCEDURE
BS DetermineSizeProc ; CALL A PROCEDURE

end
```

. *****

to BS CreatePatchStatisticsProc

```
let currentXcor 0
let currentYcor 0
let currentWho 0
let currentColor 0

create-PatchStatistics Npatches ; Npatches is set in BS AnalyseProc and equals the number of
identified food patches
[
  set size 2
```

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 shape "circle"
set currentWho Who
set patchInfo "no info"
; define the patch location by calculating the mean x and y-coordinates
; of the grdcells (Netlogo patches) that are part of this food patch:
set xcor mean [ pxcor ] of patches with [ flowerPatchID = currentWho ]
set ycor mean [ pycor ] of patches with [ flowerPatchID = currentWho ]
set currentColor patchcolor
if currentWho != flowerPatchID ; i.e. if calculated location is outside of the
; actual food patch area:

[
ask min-one-of patches with [ flowerPatchID = currentWho ] [distance myself]
]
[
set currentXcor pxcor
set currentYcor pycor
set currentColor patchcolor
]
setxy currentXcor currentYcor
]
set color currentColor - 1
set label-color white
set label who
let memoFlowerSpeciesList []
let memoPatchType -1
let patchcolorMemo patchcolor

ask habitats with [ habitatColourID = patchcolorMemo and habitatSwitchedOn? = true ]
[
set memoFlowerSpeciesList flowerspecieslist
set memoPatchType habitatType ; NOTE: this flowerspecieslist is the habitats-own variable
]
set patchType memoPatchType
set flowerSpeciesList memoFlowerSpeciesList ; NOTE: here, flowerspecieslist is a
patchstatistics-own variable
]
end

```

, *****
,

to BS DetermineSizeProc

```

let currentWho 0
foreach sort patchStatistics
[
ask ?
]
set currentWho Who
set areaPx (count patches with [flowerPatchID = currentWho])

```

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 areaSqm round(areaPx * BS_Scaling_m/NLpatches * BS_Scaling_m/NLpatches)  
set perimeter_m BS_PerimeterREP  
]  
]  
end
```

. *****
,

to-report BS PerimeterREP

```
let whoPatch who  
let borderCells 0  
  
ask patches with [ flowerPatchID = whoPatch ]  
[  
let pcolorMemo patchcolor  
if count neighbors with [ patchcolor != pcolorMemo ] > 0  
[ set borderCells borderCells + 1 ]  
]  
set perimeter_m borderCells * BS_Scaling_m/NLpatches  
report perimeter_m  
end
```

. *****
,

to CreateHabitatsProc

```
ifelse file-exists? HabitatsFile ; SYSTEM_Habitats.csv (defined in Parameter.csv)  
[  
set HabitatDataCSV csv:from-file HabitatsFile ; read file & save data; the csv file contains  
information about which habitat types are represented by which colour and the relative abundance  
of flower species  
let headerList item 0 HabitatDataCSV ; the header of the csv table is saved in headerList:  
["habitatColourID" "colourRangeMin" "colourRangeMax" "habitatSwitchedOn?" "habitatType"  
"ONLY FLOWER DATA BEYOND THIS COLUMN!" "Alsike_clover" "Bugle" ...]  
let firstColumnWithFlowerspecies (position "ONLY FLOWER DATA BEYOND THIS COLUMN!"  
headerList) + 1 ; the column of the first plant species (probably Alsike_clover)  
foreach but-first HabitatDataCSV ; goes through all 'lines' (except of header) in ordered way  
[  
let currentFlowerList []  
let myHabitatType "none"  
let column 0  
let nameOfCurrentList first ?  
create-habitats 1  
]  
]  
set habitatColourID item (position "habitatColourID" headerList) ? ; searches for  
"habitatColourID" in the header to identify the column,
```


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*.

```

_____ ; then takes the value of this column for the current data
line
_____ set colourRangeMin item (position "colourRangeMin" headerList) ?
_____ set colourRangeMax item (position "colourRangeMax" headerList) ?
_____ set habitatType item (position "habitatType" headerList) ?
_____ set habitatSwitchedOn? item (position "habitatSwitchedOn?" headerList) ?

_____ foreach ? ; goes through all 'columns' of this 'line' to collect data to populate flowerspecieslist:
_____ [
_____ let content ? ; the content of a single 'cell'
_____ if column >= firstColumnWithFlowerspecies and content > 0
_____ [
_____ set content precision content 3 ; content here represents the relative abundance of a flower
species in the current habitat type
_____ let speciesNameString (word "[" "\"" item column headerList "\"")
_____ set currentFlowerList lput (word speciesNameString " " content "]" ) currentFlowerList
_____ ]
_____ set column column + 1
_____ ]
_____ set flowerspecieslist currentFlowerList
_____ if nameOfCurrentList = "FlowerSpeciesList Legume"
_____ [
_____ set FlowerSpeciesList Legume currentFlowerList
_____ if habitatType != "undefined" [ set Patchtype Legume (word "\"" habitatType "\"") ]
_____ ]

_____ if nameOfCurrentList = "FlowerSpeciesList Margin"
_____ [
_____ set FlowerSpeciesList Margin currentFlowerList
_____ if habitatType != "undefined" [ set Patchtype Margin (word "\"" habitatType "\"") ]
_____ ]

_____ if nameOfCurrentList = "FlowerSpeciesList Plot"
_____ [
_____ set FlowerSpeciesList Plot currentFlowerList
_____ if habitatType != "undefined" [ set Patchtype Plot (word "\"" habitatType "\"") ]
_____ ]
_____ ] ; end CREATE HABITATS
_____ ] ; end "foreach but-first HabitatDataCSV"
_____ ] ; end if(else) file-exists?
_____ [ ; ELSE: if file doesn't exist:
_____ user-message (word "The specified HabitatsFile file cannot be loaded: " HabitatsFile)
_____ ]
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 BS WriteBumbleBeehaveOutProc [userInput?]

```
let userName ""
ifelse userInput? = true
[
  let userNamePrelim user-input (word "Name of your map: " remove " SYSTEM " substring
  MyMap 0 (length MyMap - 4) ".png" " (leave blank to not change name or insert new name (no
  extension, no quotation marks)): ")
  if length userNamePrelim > 4 and item (length userNamePrelim - 4) userNamePrelim = "." ; item
  (length userNamePrelim - 4) defines the position of the dot, as file extension is 3 characters long
  [ set userNamePrelim substring userNamePrelim 0 (length userNamePrelim - 4) ] ; removes file
  extensions
  ifelse userNamePrelim = ""
  [ set userName substring MyMap 0 (length MyMap - 4) ]
  [ set userName userNamePrelim ]
]
[ set userName remove ".png" MyMap ]
set userName (word remove " SYSTEM " userName) ; to make sure that system files are not
overwritten!
set FoodsourcesFile (word userName " Foodsources.txt")
set BumbleBeehave Inputfile FoodsourcesFile
set MyMap (word userName ".png") ; the name of the png image file that is going to be created
ask turtles [ hide-turtle ] ; we don't want to see bees etc. on the map!
export-view (word userName ".png") ; here, the actual map image is created
if file-exists? BumbleBeehave Inputfile [ file-delete BumbleBeehave Inputfile ] ; new file will be
created - delete the old first!
file-open BumbleBeehave Inputfile
file-print BS Scaling m/NLpatches ; 1. line: only one value, the scaling
file-print count patchStatistics ; 2. line: only one value, tnumber of food patches
;; NOTE: file format was modified for Beestew, NEW: perimeter m REMOVED: quantityPollen g
proteinPollenProp quantityNectar l concentration mol/l startDay stopDay corollaDepth mm
nectarFlowerVolume myl intFlowerTime s
;; file-print "id patchType patchColour xcor ycor size sqm quantityPollen g proteinPollenProp
quantityNectar l concentration mol/l startDay stopDay corollaDepth mm nectarFlowerVolume myl
intFlowerTime s flowerSpeciesList perimeter m info"
file-print "id patchType patchColour xcor ycor size sqm flowerSpeciesList perimeter m info" ; 3.
line: header
foreach sort patchStatistics
[
  ask ? ; now the actual data for each food patch are written in the file:
  [
    file-type (who - count habitats) file-type " " ; number of habitat-turtles subtracted, to make sure,
    ID listed in foodsource text file is identical to "who" of that foodsource
    file-type "\"\"
    file-type patchType
    file-type "\"\"
    file-type " "
    file-type color + 1 file-type " "
    file-type precision xcor 3 file-type " "
    file-type precision ycor 3 file-type " "
    file-type precision areaSqm 1 file-type " "
```

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.

```
file-type flowerSpeciesList file-type " "  
file-type perimeter m file-type " "  
file-write patchInfo ; printed as string i.e. with " "  
file-print ( " "  
]  
]  
file-close  
end
```

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

to DrawProc

```
let finished? false ; set to true, when the mouse pointer leaves the map  
let startTime timer  
while [ finished? = false ] ; this loop is continuously run while the mouse pointer remains on the map  
[  
let clickedXcor mouse-xcor ; current position of the mouse  
let clickedYcor mouse-ycor  
ask turtles with [ breed != brushSigns ] [ hide-turtle ] ; hide all bees etc.  
ask brushSigns  
_____  
setxy mouse-xcor mouse-ycor ; brush is located at the location of the mouse  
set size BrushSize ; brush sign gets the right size  
ifelse CircularBrush? = true ; brush shaoe either circular or square  
[ set shape "circleline2" ] ; "circle"  
[ set shape "squareline" ] ; "square"  
show-turtle ; brush becomes visible  
if SetColour != "Restore"  
[ set color read-from-string SetColour ]  
display  
_____  
if mouse-down? ; the user starts drawing!  
[  
let currentColor grey  
if SetColour != "Restore" ; SetColour is chosen by user (button 3), "Restore" erases unsaved  
drawings  
[ set currentColor read-from-string SetColour ]  
ifelse CircularBrush? = true  
[ ; circular brush  
ask patches with [ distancexy (round clickedXcor) (round clickedYcor) <= (BrushSize / 2) ]  
_____  
ifelse SetColour = "Restore"  
[ set pcolor pcolorSave ] ; erasing  
[ set pcolor currentColor ] ; drawing  
_____  
_____  
[ ; square brush
```

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 patches with [ (pxcor >= clickedXcor - (BrushSize / 2) and pxcor <= clickedXcor + (BrushSize / 2))
and (pycor >= clickedYcor - (BrushSize / 2) and pycor <= clickedYcor + (BrushSize / 2)) ]
[
  ifelse SetColour = "Restore"
  [ set pcolor pcolorSave ] ; erasing
  [ set pcolor currentColor ] ; drawing
]
]
]
ifelse CircularBrush? = true ; calculate area covered by brush size
[ set BrushArea ha (pi * ((BrushSize / Scaling NLpatches/m) / 2) ^ 2) / 10000 ] ; circular brush
[ set BrushArea ha (BrushSize / Scaling NLpatches/m) ^ 2 / 10000 ] ; square brush
ask brushSigns [ if mouse-inside? = false [ hide-turtle ] ] ; make sure brush is hidden, once
"Modifying" is finished
if mouse-inside? = false and timer - startTime > 2 ; user has 2s to move the mouse inside the map
at the beginning. Moving it out again end "Drawing"
[ if mouse-inside? = false [ set finished? true ] ]
]; end of "while"
end

```

., *****,

to SaveLoadSettingsProc [SaveOrLoad? suffix]

```

; SaveOrLoad? can be: "SaveOrLoad!" "Save!" "Load!"
; suffix: either "" or set by user to specify map swith CSO, e.g. "margins"
(see StewardshipOptionsUpdateFoodSourcesProc)
let choice ""
if SaveOrLoad? = "SaveOrLoad!"
[ set choice user-one-of "Do you want to save the current setting for this particular map? Or would
you like to import the setting from another map?" ["Save current setting" "Import from another
map/Parameters file"] ]
if SaveOrLoad? = "Load!" [ set choice "Import from another map/Parameters file" ]
if SaveOrLoad? = "Save!" [ set choice "Save current setting" ]
if choice = "Save current setting"
[
  let header ""
  ifelse file-exists? " SYSTEM Parameters.csv"
  [
    file-open " SYSTEM Parameters.csv"
    set header csv:from-row file-read-line
    file-close
  ]
  [ user-message "Can't find input file ' SYSTEM Parameters.csv'" ]

  let parameterValues [] ; this will create line 2 in the parameter file (e.g.
  foreach header ; header is a list of all parameters that are defined in the parameter file

```

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*.

```
____ ; i.e. AbundanceBoost BeeSpeciesInitialQueensListAsString Backgroundcolour...
_]
  let nextCommand (word "set MyValue " ? ) ; problem here ? refers to a string but my actually
  represent either a string or a number
  run nextCommand ; runs a command, e.g. set MaxHibernatingQueens 10000
  ifelse is-string? MyValue = true ; if string, "" need to be added: ____ (MyValue: global, as "run"
  cannot access local variable)
  [ set parameterValues lput (word "\"" MyValue "\"") parameterValues ]
  [ set parameterValues lput MyValue parameterValues ]
_]
let newParametersList []
set newParametersList lput header newParametersList
set newParametersList lput parameterValues newParametersList

if behaviorspace-run-number = 0 ; if run via BehaviorSpace then behaviorspace-run-number is > 0
and MyParametersFile is not renamed here! This is important, as a specific
____ ; parameter file needs to be created for each BehavSpace run (see
SetupBehaviorSpace)
_]
set MyParametersFile (word remove " SYSTEM " MyMap) ; make sure, system files cannot be
overwritten!
if item (length MyParametersFile - 4) MyParametersFile = "." [ set MyParametersFile substring
MyParametersFile 0 (length MyParametersFile - 4)] ; removes file extensions
set MyParametersFile (word remove suffix MyParametersFile)
set MyParametersFile (word MyParametersFile suffix " Parameters.csv")
_]
csv:to-file MyParametersFile newParametersList
_]

if choice = "Import from another map/Parameters file" ; imports the parameter file (but not the
map) from another project
_]
let myChosenParameterfile user-file ; reports the complete path - however, only the filename is
required!
ifelse member? " Parameters.csv" myChosenParameterfile ; the chosen file is a parameter file
_]
set myChosenParameterfile reverse myChosenParameterfile ;; as position reports only the first
(and not the last) occurrence, string is reversed
let cutHere position "\\" myChosenParameterfile ; this is the filename + extension, but reversed
set myChosenParameterfile substring myChosenParameterfile 0 cutHere ; reversed filename
without extension
set myChosenParameterfile reverse myChosenParameterfile ; this is the actual filename
_]
_]
____ ; the chosen file is not a parameter file but a map or foodsources file
set myChosenParameterfile reverse myChosenParameterfile ;; as position reports only the first
(and not the last) occurrence, string is reversed
let cutHere position "\\" myChosenParameterfile ; this is the filename + extension, but reversed
set myChosenParameterfile substring myChosenParameterfile 4 cutHere ; reversed filename
without extension
```

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 myChosenParameterfile reverse myChosenParameterfile ; this is the actual filename
set myChosenParameterfile remove " Foodsources" myChosenParameterfile ; this is the correct
map name
set myChosenParameterfile (word myChosenParameterfile suffix " Parameters.csv")
_]
; myChosenParameterfile has now the correct parameter file name
ifelse file-exists? myChosenParameterfile
_]
let memoOwnParameterfile? ProjectsOwnParameterFile?
set ProjectsOwnParameterFile? false ; needs to be false to load a different parameters file
set MyParametersFile myChosenParameterfile
user-message (word "Load this parameter file: " MyParametersFile)
ReadAllParametersProc ; the new parameter settings are loaded
Setup
set ProjectsOwnParameterFile? memoOwnParameterfile?
_]
[ user-message "No Parameter file is linked to the chosen map!" ]
]
end
```

. *****

to ButtonDisplayButtonsProc

```
let buttonLabelsAndCommandsList
[
[ "DISPLAY OPTIONS" "" ]
[ "ShowCohorts" "set ShowCohorts? " ]
[ "ShowDeadCols" "set ShowDeadCols? " ]
[ "ShowFoodsources" "set ShowFoodsources? " ]
[ "ShowGrid" "set ShowGrid? " ]
[ "ShowInspectedColony" "set ShowInspectedColony? " ]
[ "ShowMasterpatchesOnly" "set ShowMasterpatchesOnly? " ]
[ "ShowNests" "set ShowNests? " ]
[ "ShowPlots" "set ShowPlots? " ]
[ "ShowQueens" "set ShowQueens? " ]
[ "ShowSearchingQueens" "set ShowSearchingQueens? " ]
[ "ShowWeather" "set ShowWeather? " ]
]
VirtualButtonsProc buttonLabelsAndCommandsList
end
```

. *****

to ButtonAdvancedSetupOptionsProc

```
let buttonLabelsAndCommandsList
```

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.

```
[  
  ["ADVANCED SETUP OPTIONS" "" ]  
  ["KeepDeadColonies" "set KeepDeadColonies? " ]  
  ["MergeHedges" "set MergeHedges? " ]  
  ["MinSizeFoodSources" "set MinSizeFoodSources? " ]  
  ["RemoveEmptyFoodSources" "set RemoveEmptyFoodSources? " ]  
  ["SexLocus" "set SexLocus? " ]  
  ["StopExtinct" "set StopExtinct? " ]  
  ["UnlimitedMales" "set UnlimitedMales? " ]  
]  
VirtualButtonsProc buttonLabelsAndCommandsList  
end
```

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

to-report BS SwitchValueButtonREP [command]

```
set command remove "set " command  
if command = "" [ set command "false" ]  
set command (word "set SwitchOn " command) ; run cannot set local variables to a new value!  
run command  
report SwitchOn  
end
```

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

to VirtualButtonsProc [buttonLabelsAndCommandsList]

```
ask patches [ set pcolor 9 ]  
ask turtles [ hide-turtle ]  
let buttonXinit min-pxcor + 60  
let buttonYinit max-pycor - 16  
let buttonX buttonXinit  
let buttonY buttonYinit  
let labelXshift -10  
let labelYshift -2  
let buttonsYdistance 8  
let buttonsXdistance 20  
let headerShift x 20  
let leave? false  
let save? false  
let apply? false  
  
foreach buttonLabelsAndCommandsList  
[  
  create-buttons 1 ; the BUTTON  
  [  
    setxy buttonX buttonY
```

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 size 7
set color 7
set myLabel item 0 ?
set myCommand item 1 ?
set exitButton? false
set headerButton? false
if myCommand = ""
[ set headerButton? true
hide-turtle ]
set on? BS SwitchValueButtonREP myCommand
ifelse on? = true
[ set shape "switchButtonOn" ]
[ set shape "switchButtonOff" ]
]
create-buttonLabels 1 ; the LABEL
[
setxy buttonX + labelXshift buttonY + labelYshift
set shape "invisible"
set color grey
set size 0.1
set label-color black
set label item 0 ?
if item 1 ? = ""
[ set xcor xcor + headerShift_x ]
]
set buttonY buttonY - buttonsYdistance
if buttonY < 1.5 * buttonsYdistance
[
set buttonX buttonX + labelXshift
set buttonY buttonYinit
]
]
create-buttons 1 ; the APPLY & SAVE BUTTON
[
setxy max-pxcor - 20 max-pycor - 20
set shape "circle"
set size 12
set color green
set myLabel "APPLY & SAVE"
set exitButton? true
set on? false
]

create-buttonLabels 1 ; the LABEL
[
setxy (max-pxcor - 20 + labelXshift) (max-pycor - 20 + labelYshift)
set shape "invisible"
set color grey
set size 0.1
set label-color black
```


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 label "APPLY & SAVE"
]

create-buttons 1 ; the APPLY BUTTON
[
setxy max-pxcor - 20 max-pycor - 40
set shape "circle"
set size 12
set color green
set myLabel "APPLY"
set exitButton? true
set on? false
]
create-buttonLabels 1 ; the LABEL
[
setxy (max-pxcor - 20 + labelXshift) (max-pycor - 40 + labelYshift)
set shape "invisible"
set color grey
set size 0.1
set label-color black
set label "APPLY"
]

create-buttons 1 ; the CANCEL BUTTON
[
setxy max-pxcor - 20 max-pycor - 60
set shape "circle"
set size 12
set color red
set myLabel "CANCEL"
set exitButton? true
set on? false
]
create-buttonLabels 1 ; the LABEL
[
setxy (max-pxcor - 20 + labelXshift) (max-pycor - 60 + labelYshift)
set shape "invisible"
set color grey
set size 0.1
set label-color black
set label "CANCEL"
]

while [ leave? = false ]
[
if mouse-down?
[
let chosenButton nobody
ask patch mouse-xcor mouse-ycor
]
```

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 count buttons with [ distance myself < (buttonsYdistance / 2) ] > 0
    [ set chosenButton [who] of one-of buttons with [ distance myself < (buttonsYdistance / 2) ] ]
  ]
  if chosenButton != nobody
  [
    ask button chosenButton
  ]
  ifelse on? = false
  [ set on? true set shape "switchButtonOn" ]
  [ set on? false set shape "switchButtonOff" ]
  ]
  display
  wait 0.2
  ]
  ]

  ask buttons with [ myLabel = "APPLY & SAVE" ]
  [
    if on? = true
    [
      set leave? true
      set save? true
    ]
  ]

  ask buttons with [ myLabel = "APPLY" ]
  [
    if on? = true
    [
      set leave? true
      set apply? true
    ]
  ]

  ask buttons with [ myLabel = "CANCEL" ]
  [
    if on? = true
    [
      set leave? true
      set save? false
    ]
  ]
  ] ; while leave? false

  if save? = true or apply? = true
  [
    let commandList []
    foreach sort-on [ who ] buttons
    [ ask ?
  ]
  ]
```

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 on? = true
    [
        if myCommand != "" and myCommand != 0
        [
            set myCommand (word myCommand "true")
            set commandList lput myCommand commandList
        ]
    ]
    [
        if myCommand != "" and myCommand != 0
        [
            set myCommand (word myCommand "false")
            set commandList lput myCommand commandList
        ]
    ]
    [
        foreach commandList [ run ? ]
    ]
    ask buttons [ die ]
    ask buttonLabels [ die ]
    UpdateViewProc
    if save? = true [ SaveLoadSettingsProc "Save!" "" ]
end
```

. *****

to UpdateViewProc

```

ask deadCols [ set color white set size 1 set label "" ]
ask foodsources [ set color colorMemo ]
ask Colonies [ set label ColonySize ]
ifelse showFoodSources?
[
    ifelse showMasterPatchesOnly?
    [
        ask Foodsources with [masterPatch?] [show-turtle]
        ask Foodsources with [not masterPatch?] [hide-turtle]
    ]
    [ ask Foodsources [show-turtle]]
]
[ ask foodsources [hide-turtle]]
ifelse showSearchingQueens?
[ ask bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ] [show-turtle]]
[ ask bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ] [hide-turtle]]
ifelse showCohorts?
[ ask bees with [ shape = "halfline" ] [show-turtle]]
[ ask bees with [ shape = "halfline" ] [hide-turtle]]
ifelse showQueens?
```

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 with [ caste = "queen" and mated? = true ] [show-turtle]]
[ ask bees with [ caste = "queen" and mated? = true ] [hide-turtle]]
ifelse showNests?
[ ask colonies [show-turtle]]
[ ask colonies [hide-turtle]]
ifelse showDeadCols?
[ ask deadCols [show-turtle]]
[ ask deadCols [hide-turtle]]
ask badgers [ show-turtle]
let labelPatch patch 295 5
ifelse showGrid?
[
  ask patches [ set pcolor pcolorSave ]
  ask patches with [ remainder pxcor round (gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor black ]
]
ask patches with [ remainder pycor round (gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor black ]
ask labelPatch [ set plabel-color black set plabel word gridsize " m" ]
]
[
  ask patches [ set pcolor pcolorSave ]
  ask labelPatch [ set plabel "" ]
]
end
```

```
. *****
/
```

to ButtonBackgroundImageProc

```
user-message "Select an image file to be used as background image"
set BackgroundImage user-file
import-pcolors BackgroundImage
ask patches [ set satelliteColor pcolor ]
end
```

```
... *****
//
```

to-report FilenameREP [extension?]

```
let path user-file
set path reverse path ;; as position reports only the first (and not the last) occurrence, string is reversed
let cutHere1 0
if extension? = false [ set cutHere1 1 + position "." path ]
let cutHere2 position "\\\" path ; this is the filename + extension, but reversed
set path substring path cutHere1 cutHere2 ; reversed filename without extension
let myFilename reverse path ; this is the actual filename
report myFilename
```

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 ButtonSelectFieldProc

```
let finished? false  
ask signs with [ shape = "circletarget" ] [ show-turtle ] ; pointer becomes visible  
let startTime timer  
while [ finished? = false ]  
[  
  let clickedXcor mouse-xcor ; updated continuously!  
  let clickedYcor mouse-ycor  
  ask signs with [ shape = "circletarget" ] [ setxy mouse-xcor mouse-ycor ] ; pointer is following the  
  mouse  
  if mouse-down?  
  [  
    ask min-one-of foodsources with [ masterpatch? = true ] [ distancexy clickedXcor clickedYcor ] ;  
    closest masterpatch is selected  
    [  
      ifelse member? who SelectedFieldsList = false ; if the masterpatch ID of the selected field is not  
      already in the list of selected fields..  
      [  
        ;if member? "Crop" patchType ; and the field is a crop  
        ifelse member? "Crop" patchType and length flowerspeciesList = 1 ; and the field is a crop  
        without a CSO already applied  
        [  
          set SelectedFieldsList lput who SelectedFieldsList ; then the masterpatch ID is added to the  
          list of selected fields  
          let memoWho who  
          ask foodsources with [ masterpatchID = memoWho ]  
          [  
            ifelse member? ["plot"] flowerspeciesList or member? ["margin"] flowerspeciesList or  
            member? ["legume"] flowerspeciesList  
            [ set shape "csodeselected" ] ; mark a field to remove the CSO  
            [ set shape "csoselected" ] ; mark a selected field  
          ]  
          wait 0.2  
        ]  
        [ user-message "Stewardship options can only be applied to crop fields" ]  
      ]  
      [ ; if it had been selected already, it is now de-selected!  
        set SelectedFieldsList remove who SelectedFieldsList  
        let memoWho who  
        ask foodsources with [ masterpatchID = memoWho ] [ set shape "circle" ] ; gets back its  
        original shape  
        wait 0.2  
      ]  
    ]  
  ]
```

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 mouse-inside? = false and timer - startTime > 2 ; user has 2s to move the pointer on the map
_]
  ask signs with [ shape = "circletarget" ] [ hide-turtle ]
  if mouse-inside? = false [ set finished? true ] ; if pointer is moved outside, loop is stopped
_]
_]
end
```

```

. *****
,
```

to ButtonStewardshipOptionsProc

; UK Mid-Tier Countryside Stewardship Options: legume: AB15, margin: AB8, plot: AB1
CreateHabitatsProc ; habitat needed to access their data (flower lists/abundances). Will die in the
subsequent Setup

```

let marginWidth m 0
let marginLength m 0
let plotArea m2 0
let cssArea m2 0
let cssFlowerspeciesList []
```

```

if StewardshipOption = "margin"
[
  set marginWidth m read-from-string user-input "How wide [m] is your margin?"
  set marginLength m read-from-string user-input "How long [m] is your margin?"
  set cssFlowerspeciesList FlowerSpeciesList Margin ; flower composition and abundance in this
CSO
_]

```

; get the CSO areas:

```

if StewardshipOption = "plot"
[
  set plotArea m2 10000 * read-from-string user-input "Which area [ha] would you like to set
aside?"
  set cssFlowerspeciesList FlowerSpeciesList Plot ; flower composition and abundance in this CSO
_]

```

if StewardshipOption = "legume"

```

[ set cssFlowerspeciesList FlowerSpeciesList Legume ] ; flower composition and abundance in this
CSO

```

; apply CSOs to selected fields:

```

foreach SelectedFieldsList
[
  let cssSpeciesList []
  let cssAbundanceList []

```

; get the flower species & abundances:

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*.

```

foreach cssFlowerspeciesList ;; goes through all items (item e.g. ["Bugle" 0]) e.g. ;
cssFlowerspeciesList (Plot): ["Alsike clover" 24.727] ["Bugle" 0] ["Burdock" 0] ["Oilseed rape" 0]
["Giant bindweed" 0] ["Common knapweed" 9.833] ["Greater knapweed" 0]
["Rosebay willowherb" 0] ["Marsh thistle" 0] ["Spear thistle" 0] ["Hawthorn" 0] ["Foxglove" 0]
["Wild teasel" 0] ["Vipers bugloss" 0] ["Ground ivy" 0] ["Bluebell" 0] ["St Johns wort" 0]
["Field scabious" 0] ["White dead nettle" 0] ["Red dead nettle" 0] ["Birdsfoot trefoil" 37.333]
["Selfheal" 3.5] ["Blackthorn" 0] ["Buttercup" 0] ["Dog rose" 0] ["Bramble" 0] ["Average Willow" 0]
["Ragwort" 0] ["Hedge woundwort" 0] ["Comfry" 0] ["Dandelion" 0] ["Red clover" 3.737]
["White clover" 0] ["Tufted vetch" 27.8] ["Common vetch" 0] ["Crop Field beans" 0]
["Crop Oilseed rape" 0] ["Crop Maize" 0]
]
let nextItem read-from-string ? ; e.g. the item ["Common vetch" 1] is made into the list
[Common vetch 1], i.e. loses the double quotes
set cssSpeciesList lput item 0 nextItem cssSpeciesList ; e.g. [Alsike clover Common knapweed
Spear thistle ..]
set cssAbundanceList lput item 1 nextItem cssAbundanceList ; e.g. [10 14 10..]
]
ask foodsource ?
]
set stewardshipSpeciesList lput StewardshipOption stewardshipSpeciesList ; StewardshipOption:
"margin" "plot" or "legume"
let i 0
; calculate CSO areas:
if StewardshipOption = "margin"
[ set cssArea m2 marginWidth m * marginLength m ]
if StewardshipOption = "plot"
[ set cssArea m2 plotArea m2 ]
if StewardshipOption = "legume"
[ set cssArea m2 area sqm ]
if cssArea m2 > area sqm
]
user-message "Area for stewardship options of one or more fields larger than the actual field!"
set cssArea m2 area sqm
]

foreach cssSpeciesList ; e.g. cssSpeciesList: [Alsike clover Bugle Burdock Oilseed rape
Giant bindweed Common knapweed Greater knapweed Rosebay willowherb Marsh thistle
Spear thistle Hawthorn Foxglove Wild teasel Vipers bugloss Ground ivy Bluebell St Johns wort
Field scabious White dead nettle Red dead nettle Birdsfoot trefoil Selfheal Blackthorn Buttercup
Dog rose Bramble Average Willow Ragwort Hedge woundwort Comfry Dandelion Red clover
White clover Tufted vetch Common vetch Crop Field beans Crop Oilseed rape Crop Maize]
]
let currentAbundance item i cssAbundanceList ; this describes the flower density of margin
species in the margin. However, total patch/field area is larger, hence density must be reduced:
set currentAbundance precision (currentAbundance * (cssArea m2 / area sqm)) 3
set patchInfo (word cssArea m2)
set stewardshipSpeciesList lput (word "[" ? " " currentAbundance ")") stewardshipSpeciesList
set i i + 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*.

```

; now the flower abundance in the actual patch needs to be adjusted/reduced:
let abundanceOld item 1 flowerspecies relativeAbundanceList ; e.g. [Crop Field beans 107.5]
let abundanceNew precision (abundanceOld * (1 - (cssArea m2 / area sqm))) 3
set flowerspecies relativeAbundanceList but-last flowerspecies relativeAbundanceList ; removes
last item, i.e. abundance value
set flowerspecies relativeAbundanceList lput abundanceNew
flowerspecies relativeAbundanceList ; adds the new abundance value
] ; end ask foodsource
]
StewardshipOptionsUpdateFoodsourcesProc TRUE ; userInput? true
set SelectedFieldsList []
end

```

. *****

to StewardshipOptionsUpdateFoodsourcesProc [userInput?]

; creates a new FoodsourcesFile "x Foodsources.txt", in which the new stewardship options are taken into account

;; UPDATE THE FOODSOURCES INPUT FILE:

set FoodsourcesFile remove " SYSTEM " FoodsourcesFile ; "System" files cannot be overwritten
let suffix user-input "Optionally: Provide a suffix for this stewardship scenario (e.g. 'margins1') to avoid overwriting of your current files - or just click ok: "

set MyMap remove " SYSTEM " MyMap

set MyMap remove ".png" MyMap

ifelse suffix = ""

[set MyMap (word MyMap ".png")]

[set MyMap (word MyMap " " suffix ".png")]

ask turtles [hide-turtle]

export-view MyMap ; exports the view to create an image file of the current map.

file-close ; no file should be open here

set MyParametersFile remove " SYSTEM " MyParametersFile

set FoodsourcesFile remove " SYSTEM " FoodsourcesFile

set FoodsourcesFile remove " Foodsources" FoodsourcesFile

set FoodsourcesFile remove ".txt" FoodsourcesFile

ifelse suffix = ""

[set FoodsourcesFile (word FoodsourcesFile " Foodsources.txt")]

[set FoodsourcesFile (word FoodsourcesFile " " suffix " Foodsources.txt")]

; the old input file is deleted, a new one is created:

if file-exists? FoodsourcesFile [file-delete FoodsourcesFile]

file-open FoodsourcesFile

file-print 1 / Scaling NLpatches/m ; equivalent to BS Scaling m/NLpatches

file-print count foodsources with [masterpatch? = true]

file-print "id patchType patchColour xcor ycor size sqm flowerSpeciesList perimeter m info"

; METHOD: first: address all flowerpatches, write the patch data (like size, location etc.) in the new file, then address all "layers" of that 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*.

```
; layers can occur ("occurrence") in the 'patch' 'margin' 'legume', 'plot' (or 'blueberryPlot') (e.g.
occurrence = "margin")
foreach sort foodsources with [ masterpatch? = true ]
[
  ask ?
]
[
  file-type who file-type " " ; column 1
  file-type "\"
  file-type patchType      ; column 2
  file-type "\"
  file-type " "
  file-type color + 1 file-type " " ; column 3
  file-type precision xcor 3 file-type " " ; ; column 4
  file-type precision ycor 3 file-type " " ; column 5
  file-type precision area_sqm 1 file-type " " ; column 6

  let memoWho who
  let flowerSpeciesListNew [ ] ; this is to re-create flowerspeciesList but with protected quotation
  marks
  let occurrenceList fput "patch" CSS OptionsList ; CSS OptionsList [ "margin" "legume" "plot" ]
  let added? false

  foreach sort foodsources with [ masterpatchID = memoWho ] ; all layers of the currently
  addressed masterpatch are asked..
  [
    ask ?
  ]
  [
    ; occurrenceList = [patch margin legume plot (blueberryPlot)]
    foreach occurrenceList ; .. first asking those originally in the patch and then ordered by the
    CSO option: ; occurrenceList = [patch margin legume plot (blueberryPlot)]
  ]
  [
    if any? foodsources with [ masterpatchID = memoWho and occurrence = ? and occurrence !=
    "patch" ] ; adds 'CSO-marker': "legume" "plot" or "margin" to the list to specify that the following
    flowerspecies are not part of the original field but of the CSO
  ]
  [
    if added? = false ; make sure it is added only once!
    [ set flowerSpeciesListNew lput (word "[" "\" ? "\" "\"") flowerSpeciesListNew ]
    set added? true
  ]

  if occurrence = ? ; adds the actual flower species and their abundances:
  [
    set flowerSpeciesListNew lput (word "[" "\" (item 0 flowerspecies relativeAbundanceList)
    "\" " " (item 1 flowerspecies relativeAbundanceList) "]" ) flowerSpeciesListNew
    let i 0
    foreach stewardshipSpeciesList ; [ ] or e.g. [plot [Common knapweed 0.166]
    [Greater knapweed 0.166] ...]
  ]
  [
    ifelse member? ? CSS OptionsList
  ]
  [
```

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 flowerSpeciesListNew lput (word "[" "\" ? "\"") flowerSpeciesListNew ; this adds e.g.
    "plot" to flowerSpeciesListNew to indicate that the following flower species are present in the CSO
    (plot)

    ; here: flowerSpeciesListNew e.g.
    ["[" "Crop Oilseed rape\" 556.763]" "\"plot\""]

    set i i + 1
  ]
  [ ; adding protected quotation marks:
    if length flowerSpeciesList = 1 ; to make sure CSO/blueberry patches are not repeatedly
    added
    [
      let newSpeciesData item i stewardshipSpeciesList ; e.g. [MarginBirdsfoot trefoil 1000]
      let blankPosition position " " newSpeciesData
      let newSpecies substring newSpeciesData 1 blankPosition ; position 0 is "[" hence start at
      position 1
      let newSpeciesAbundance substring newSpeciesData blankPosition (length
      newSpeciesData - 1) ; last position is "]" hence -1
      set flowerSpeciesListNew lput (word "[" "\" newSpecies "\" " " newSpeciesAbundance
      "]" ) flowerSpeciesListNew
      set i i + 1
    ]
  ]
  ] ; end if occurrence
] ; end foreach occurrenceList
] ; end ask ?
] ; end "foreach sort foodsources"
if length flowerSpeciesList > 1 and member? "Crop" patchType and member? who
SelectedFieldsList
[ set flowerSpeciesListNew (word "[" "[" "\" (item 0 flowerspecies relativeAbundanceList) "\" " "
(item 1 flowerspecies relativeAbundanceList) "]" ) ]
file-type flowerSpeciesListNew file-type " " ; ; column 7: flowerSpeciesList
file-type perimeter_m file-type " " ; ; column 8
set patchInfo remove "\" patchInfo ; to avoid accumulation of \
file-write (word patchInfo) ; ; column 8 ; "no info" ; printed as string i.e. with " "
file-print ( " ")
]
]
file-close

SaveLoadSettingsProc "Save!" (word " " suffix)
Setup

if StewardshipOption = "margin"
[
  ask foodsources with [ shape = "fieldmargin" ]
]
let memoMasterpatch masterpatchID
ask foodsources with [ masterpatchID = memoMasterpatch ] [ set shape "fieldmargin" ]
]

```

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 StewardshipOption = "plot"
[
ask foodsources with [ shape = "fieldplot" ]
[
let memoMasterpatch masterpatchID
ask foodsources with [ masterpatchID = memoMasterpatch] [ set shape "fieldplot" ]
]
]
]
if StewardshipOption = "legume"
[
ask foodsources with [ shape = "fieldlegume" ]
[
let memoMasterpatch masterpatchID
ask foodsources with [ masterpatchID = memoMasterpatch] [ set shape "fieldlegume" ]
]
]
end
```

. *****

to StewardshipAreasProc

```
let farmArea ha read-from-string user-input "What is your total farm area (hectare)?"
output-print "Total stewardship area: " output-print (word precision CSS TotalAreaAll ha 2 " ha ("
precision (100 * CSS TotalAreaAll ha / farmArea ha) 1 " % of farm area)") output-print ""
output-print "Margins: " output-print (word precision CSS TotalAreaMargin ha 2 " ha (" precision
(100 * CSS TotalAreaMargin ha / farmArea ha) 1 " % of farm area)") output-print ""
output-print "Legume fields: " output-print (word precision CSS TotalAreaLegume ha 2 " ha ("
precision (100 * CSS TotalAreaLegume ha / farmArea ha) 1 " % of farm area)") output-print ""
output-print "Plots: " output-print (word precision CSS TotalAreaPlot ha 2 " ha (" precision (100 *
CSS TotalAreaPlot ha / farmArea ha) 1 " % of farm area)") output-print ""
end
```

. *****

to ButtonGenerateOutputProc

```
set Report name ""
let replicates 20 ; read-from-string user-input "How many replicates would you like to run (e.g.
10)?"
let timesteps 365 * 5 ; read-from-string user-input "How many days would you like to simulate in
each run (e.g. 365)?"
let reportType user-one-of "Choose whether to run a full simulation (may take several hours) or
just a quick estimate on colony numbers?" ["Full simulation" "Resources provided"]
if reportType = "Resources provided"
[
let totalPollenOnMapPerYear kg 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*.

```

let totalNectarOnMapPerYear l 0
ask foodsources
[
  if stopDay < startDay [ user-message "Foodsource stops flowering before it has even started!" ]
  let pollenAdded kg pollenMax g * (stopDay - startDay) / 1000
  let nectarAdded l nectarMax myl * (stopDay - startDay) / (1000 * 1000)
  set totalPollenOnMapPerYear kg totalPollenOnMapPerYear kg + pollenAdded kg
  set totalNectarOnMapPerYear l totalNectarOnMapPerYear l + nectarAdded l
]

; Rotheray et al 2017, J Apic Res, Vol. 56, No. 3, 288–299: colonies require 176 g pollen and 1186 g
sugar in their lifetime
let nColonies pollen Rotheray totalPollenOnMapPerYear kg * 1000 / 176
let totalSugar kg totalNectarOnMapPerYear l * 0.342 ; assuming sucrose concentration of 1mol/l
and 0.342 kg sucrose per mol
let nColonies nectar Rotheray totalSugar kg / 1.186
ifelse nColonies pollen Rotheray > nColonies nectar Rotheray
[ user-message (word "Total amount of sugar per year: " round totalSugar kg "kg. Total amount
of pollen per year: " round totalPollenOnMapPerYear kg "kg. The bee population is limited by lack of
nectar! The maximal number of colonies supported is up to ca. " round nColonies nectar Rotheray)
]
[ user-message (word "Total amount of sugar per year: " round totalSugar kg "kg. Total amount
of pollen per year: " round totalPollenOnMapPerYear kg "kg. The bee population is limited by lack of
pollen! The maximal number of colonies supported is up to ca. " round nColonies pollen Rotheray)
]
]

if reportType = "Full simulation"
[
  set Report_name (word " " user-input "What is the name of your results?")
  let reportNameSave Report_name ; Report_name needs to be a global variable to be accessible in
CreateFoodsourcesProc, but also local, to not be deleted by Setup!
  if file-exists? ( word Report_name ".csv") [ file-delete ( word Report_name ".csv") ]
  user-message (word "Your results named " Report_name " are about to be created - this may take
a while! To speed things up, set the 'slider' to faster and untick the 'view updates' box. The progress
of the simulation is shown in the 'Output' window")
  file-open ( word Report_name ".csv")
  file-type "Timestep," file-type "Colonies," file-type "Pollinators," file-type remove ".csv"
Report_name file-type "," file-type replicates file-type "," file-type timesteps file-print ","
  set RAND_SEED 1
  repeat replicates
  [
    Setup
    set Report_name reportNameSave ; needs to be re-set after "clear-all"
    output-print (word "Progress of the simulation: " precision (100 * (RAND_SEED / replicates)) 1 "
%)
  ]

  file-open ( word Report_name ".csv")
  repeat timesteps

```

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*.

```

__|
__|   Go
__|   file-type ticks file-type " ," file-type TotalColonies file-type " ," file-print TotalAdultworkers
__|   if AssertionViolated = true
__|   |
__|   |   ask patches [ set pcolor red ]
__|   |   stop
__|   |
__|   |
__|   |   file-close
__|   |   set RAND_SEED RAND_SEED + 1
__|   |
__|   |   user-message (word "Your results named " Report_name " are now finished! Create a report using
the My BEE-STEWARD Report.xlsm file")
__|   |
__|   end
. *****
/

```

to ButtonChooseColourProc

```

let textUser "Choose a colour: " ; informs the user which habitat type (if any) is represented by each
colour (only if habitat is already represented on map)
carefully
__| [ set textUser (word textUser " Pink: " [ patchtype ] of one-of foodsources with [ color = Pink - 1 ])]
__| [ set textUser (word textUser " Pink: undefined")]
carefully
__| [ set textUser (word textUser "; Magenta: " [ patchtype ] of one-of foodsources with [ color =
Magenta - 1 ])]
__| [ set textUser (word textUser "; Magenta: undefined")]
carefully
__| [ set textUser (word textUser "; Brown: " [ patchtype ] of one-of foodsources with [ color = Brown -
1 ])]
__| [ set textUser (word textUser "; Brown: undefined")]
carefully
__| [ set textUser (word textUser "; Blue: " [ patchtype ] of one-of foodsources with [ color = Blue - 1
])]
__| [ set textUser (word textUser "; Blue: undefined")]
carefully
__| [ set textUser (word textUser "; Green: " [ patchtype ] of one-of foodsources with [ color = Green -
1 ])]
__| [ set textUser (word textUser "; Green: undefined")]
carefully
__| [ set textUser (word textUser "; Yellow: " [ patchtype ] of one-of foodsources with [ color = Yellow -
1 ])]
__| [ set textUser (word textUser "; Yellow: undefined")]
carefully
__| [ set textUser (word textUser "; Red: " [ patchtype ] of one-of foodsources with [ color = Red - 1 ])]
__| [ set textUser (word textUser "; Red: undefined")]

```

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*.

```
carefully
  [ set textUser (word textUser "; Turquoise: " [ patchtype ] of one-of foodsources with [ color =
Turquoise - 1 ])]
  [ set textUser (word textUser "; Turquoise: undefined")]
carefully
  [ set textUser (word textUser "; Violet: " [ patchtype ] of one-of foodsources with [ color = Violet -
1 ])]
  [ set textUser (word textUser "; Violet: undefined")]
set SetColour user-one-of textUser
[
  "Pink"
  "Magenta"
  "Brown"
  "Blue"
  "Green"
  "Yellow"
  "Red"
  "Turquoise"
  "Violet"
  "Orange"
  "Lime"
  "Cyan"
  "Sky"
  "Grey"
  "Black"
  "White"
  "Restore"
  "Define by number"
]
if SetColour = "Define by number" [ set SetColour user-input "Please define a colour, using the
NetLogo colour scheme [0..140]" ]
set Button1Monitor (word "Choose Colour (" SetColour ")")
DrawProc
end
```

. *****
,

to ButtonDisplayProc [displayOption]

```
; nectarVisits pollenVisits defaultView
let averageCumulVisits 0
let displayColor red
if displayOption = "nectarVisits" or displayOption = "pollenVisits"
[
  ask foodsources
  [
    show-turtle
    if displayOption = "nectarVisits"
    [
```

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 averageCumulVisits averageCumulVisits + cumulNectarVisits
    set displayColor yellow
  ]
  if displayOption = "pollenVisits"
  [
    set averageCumulVisits averageCumulVisits + cumulPollenVisits
    set displayColor orange
  ]
]
set averageCumulVisits averageCumulVisits / count foodsources with [ masterpatch? = true ]
ask foodsources with [ masterpatch? = true ]
[
  let myMasterpatch who
  let summedCumulVisits 0 ; all pollen or nectar visits within a layergroup
  ask foodsources with [ masterpatchID = myMasterpatch ]
]
[
  if displayOption = "nectarVisits" [ set summedCumulVisits summedCumulVisits +
cumulNectarVisits ]
  if displayOption = "pollenVisits" [ set summedCumulVisits summedCumulVisits +
cumulPollenVisits ]
]
let displayVisits summedCumulVisits / averageCumulVisits
ask foodsources with [ masterpatchID = myMasterpatch ]
[
  set color scale-color displayColor sqrt displayVisits 0 4
  ifelse round (displayVisits * averageCumulVisits) > 1
    [ set label-color 103 set label round (displayVisits * averageCumulVisits) ]
    [ set label-color 103 set label "" ]
]
]
]
if displayOption = "defaultView"
[
  ask deadCols [ set color white set size 1 set label "" ]
  ask foodsources [ set color colorMemo ]
  ask Colonies [ set label ColonySize ]
  ifelse showFoodSources?
  [
    ifelse showMasterPatchesOnly?
    [
      ask Foodsources with [masterPatch?] [show-turtle set label "" ]
      ask Foodsources with [not masterPatch?] [hide-turtle]
    ]
    [ ask Foodsources [st set label "" ] ]
  ]
  [ ask foodsources [ht] ]

  ifelse showSearchingQueens?
  [ ask bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ] [st] ]
  [ ask bees with [ caste = "queen" and colonyID = -1 and activity != "hibernate" ] [ht] ]

```

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 showCohorts?  
[ ask bees with [ shape = "halfline" ] [st] ]  
[ ask bees with [ shape = "halfline" ] [ht] ]  
  
ifelse showQueens?  
[ ask bees with [ caste = "queen" and mated? = true ] [st] ]  
[ ask bees with [ caste = "queen" and mated? = true ] [ht] ]  
  
ifelse showNests?  
[ ask colonies [st] ]  
[ ask colonies [ht] ]  
  
ifelse showDeadCols?  
[ ask deadCols [st] ]  
[ ask deadCols [ht] ]  
  
ask badgers [ st]  
  
ifelse showGrid?  
[  
ask patches with [ remainder pxcor round (gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor  
black ]  
ask patches with [ remainder pycor round (gridsize * Scaling_NLpatches/m) = 0 ] [ set pcolor  
black ]  
ask patch 290 5 [ set plabel-color black set plabel word gridsize " m" ]  
]  
[  
ask patches [ set pcolor pcolorSave ]  
ask patch 290 5 [ set plabel "" ]  
]  
]  
end  
  
. *****  
,
```

to ButtonIdentifyProc

```
let finished? false  
let displayedInfo ""  
let startTime timer  
ask foodsources with [ masterpatch? = false ] [ hide-turtle ]  
create-signs 1  
[  
set shape "identifier"  
set size 20  
set color black  
]  
while [ finished? = false ]
```


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 hoveredXcor mouse-xcor
  let hoveredYcor mouse-ycor
  let turtleID -1
  ask signs with [ shape = "identifier" ] [ setxy mouse-xcor mouse-ycor show-turtle ]
  ; ask min-one-of foodsources [ distancexy hoveredXcor hoveredYcor ]
  ask min-one-of turtles with [ shape != "identifier" ] [ distancexy hoveredXcor hoveredYcor ]
  [
    ifelse distancexy hoveredXcor hoveredYcor < 5
    [
      set turtleID who
      if breed = foodsources [ set displayedInfo patchType ]
      if breed = colonies [ set displayedInfo (word shape " colony") ]
      if breed = bees [ set displayedInfo (word speciesName ": " caste " " stage) ]
      if breed = signs [ set displayedInfo shape ]
    ]
    [
      set displayedInfo ""
      set turtleID -1
    ]
  ]
  ask signs with [ shape = "identifier" ] [ set label displayedInfo ]
  if mouse-down? = true
  [
    ifelse turtleID >= 0
    [ inspect turtle turtleID ] ; opens "inspect" window of selected turtle
    [ inspect patch mouse-xcor mouse-ycor ]
  ]
  wait 0.2
  if mouse-inside? = false and timer - startTime > 2
  [
    ask signs with [ shape = "identifier" ] [ hide-turtle ]
    let outsideTime timer
    let stopOutsideCheck false
    while [ stopOutsideCheck = false ]
    [
      wait 0.5
      if mouse-inside? = true [ set stopOutsideCheck true ]
      if timer - outsideTime > 2
      [
        set stopOutsideCheck true
        set finished? true
      ]
    ]
    ask signs with [ shape = "identifier" ] [ show-turtle ]
  ]
  [
    ask signs with [ shape = "identifier" ] [ die ]
    ask foodsources [ show-turtle ]
  ]
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 ButtonBrushSizeProc

set BrushSize read-from-string user-input "Set brush size (value between 1 - 200)"
if BrushSize < 1 [set BrushSize 1]
if BrushSize > 100 [set BrushSize 200]
ifelse CircularBrush? = true ; recalculate area covered by brush:
[set BrushArea ha (pi * ((BrushSize / Scaling_NLpatches/m) / 2) ^ 2) / 10000] ; circular
[set BrushArea ha (BrushSize / Scaling_NLpatches/m) ^ 2 / 10000] ; square
set Button2Monitor (word "Set Brush Size (" BrushSize ") (ca. " precision BrushArea ha 1 "ha))"
DrawProc
end

. *****
,

to ButtonReplaceColoursProc

let color1 -999
let color2 -999
ask turtles [hide-turtle] ; hide bees, colonies etc.
ask patches [set pcolorsave pcolor] ; save the current colour of each grid cell
; creates 2 colour scales:
ask patches with [pxcor >= (max-pxcor - 10)]
[set pcolor 140 * (pycor / max-pycor)]
ask patches with [pxcor <= 10]
[
let newColor 140 * (pycor / max-pycor)
set newColor round (newColor / 10)
set pcolor (newColor * 10) + 5
]
user-message "Select the NEW colour by clicking on a patch or the colour scale"
while [color2 = -999]
[
if mouse-down?
[set color2 [pcolor] of patch mouse-xcor mouse-ycor]
display
]
ask patches [set pcolor pcolorsave] ; colour scales are no longer shown

let replaceAllPatches? user-yes-or-no? "Click 'Yes' to replace all patches or 'No' to replace only
chosen patch, then select a patch by clicking on it."
let finished? false
let patchesToBeChecked [] ; grid cells which are neighbours of a grid cell that has its colour changed
let patchesToBeCheckedUpdate []
let chosenNLpatch nobody
while [finished? = false]

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 mouse-down? and chosenNLpatch = nobody
  [
    set chosenNLpatch patch mouse-xcor mouse-ycor
    ask chosenNLpatch
  ]
  [
    set color1 pcolor ; saves the colour of the chosen grid cell
    set pcolor color2 ; the colour of this grid cell changes
    ask neighbors4 [ set patchesToBeChecked lput self patchesToBeChecked ] ; self reports the
    asking agent (in contrast to myself, which would report chosenNLpatch)
  ]
]; end mouse-down

ifelse replaceAllPatches? = true ; if true, all grid cells of the same colour as the chosen one change
their colour
[
  ask patches with [ pcolor = color1 ][ set pcolor color2 ]
  if color1 >= 0 [ set finished? true ] ; initial value of color1: -999
]
[
  foreach patchesToBeChecked ; (true) neighbors are checked (repeatedly) whether they have the
  same color
  [
    ask ?
  ]
  [
    if pcolor = color1
    [
      set pcolor color2 ; neighbours of grid cells of the same colour change now their colour
      ask neighbors4 [ set patchesToBeCheckedUpdate lput self patchesToBeCheckedUpdate ] ; self
      reports the asking agent (in contrast to myself, which would report chosenNLpatch)
    ]
  ]
]
set patchesToBeChecked patchesToBeCheckedUpdate
set patchesToBeCheckedUpdate []
if chosenNLpatch != nobody and patchesToBeChecked = [] ; no more new neighbours of the same
colour - stop the process!
[ set finished? true ]
]
if color1 = color2
[
  set finished? true ; no need to do anything in this case!
  user-message "Identical colours - no changes were made!"
]
]; end WHILE finished? = false
ask patches [ set plabel "" ]
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 ButtonScaleBarProc

```
let xpos 1 ; xy location of the left end of the scale bar ("scalebar" is only a...  
let ypos 207 ; .."half line" hence with increasing size it only extends on the right side)  
let targetSize max-pxcor / 2.5 ; 2.5: size 120 i.e. actual length: 60 (as shown line covers only the right  
side of the turtle)  
let targetLength m targetSize / SCALING NLpatches/m / 2  
let divisor (10 ^ (round (log targetLength m 10))) / 2  
let realBarLength m divisor * ceiling (targetLength m / divisor)  
ifelse count signs with [ shape = "scalebar" or shape = "scaledistance"] > 0  
[ ask signs with [ shape = "scalebar" or shape = "scaledistance" ] [ die ] ]  
[  
  create-Signs 2  
  [  
    set color black  
    set shape "scalebar"  
    set size 2 * SCALING NLpatches/m * realBarLength m  
    setxy xpos ypos  
  ]  
  ask one-of signs with [ shape = "scalebar" ]  
  [  
    set shape "scaledistance"  
    set size 1  
    setxy xpos + 13 ypos - 4  
    set label-color black  
    set label (word realBarLength m " m")  
    if realBarLength m >= 10000  
    [ set label (word (realBarLength m / 1000) " km") ]  
  ]  
]; else  
end
```

. *****
,

to ButtonInitialQueensProc

```
let queensCommandString ""  
let terrestrisOnly? user-yes-or-no? "Set queens for B. terrestris only?"  
ifelse terrestrisOnly? = true  
[  
  let nTerrestris user-input "Initial number of B. terrestris queens: "  
  set BeeSpeciesInitialQueensListAsString (word "B terrestris " nTerrestris)  
]  
[  
  foreach BeeSpeciesDefinedList  
  [  
    let addThisSpecies? user-yes-or-no? (word "Are there any " ? " queens present?")  
    ;let addThisSpecies? user-one-of (word "Are there any " ? " queens present?") ["yes" "no"  
"finish"]  
    if addThisSpecies? = true
```

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 addThisSpecies? = "yes"  
    ]  
    let queensThisSpecies read-from-string user-input (word "How many " ? " queens are present?")  
    set queensCommandString (word queensCommandString ? " " queensThisSpecies " ")  
    ]  
    ]  
    set BeeSpeciesInitialQueensListAsString queensCommandString  
    ]  
    SaveLoadSettingsProc "Save!" ""  
    Setup  
end
```

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

to PanelSettingProc

```
if Panel = "Modify Maps"  
    ]  
    let notBrushShape "Circular"  
    if CircularBrush? = true [ set notBrushShape "Square" ]  
    set Button1Monitor (word "Choose Colour (" SetColour ")")  
    set Button2Monitor (word "Set Brush Size (" BrushSize ") (ca. " precision BrushArea ha 1 "ha)")  
    set Button3Monitor (word "Switch Brush Shape to " notBrushShape)  
    set Button4Monitor "Draw on Map"  
    set Button5Monitor "Replace Colours"  
    set Button6Monitor "Clear whole Map"  
    set Button7Monitor "Update current Map"  
    ]  
  
if Panel = "Stewardship Options"  
    ]  
    set Button1Monitor (word "Select Stewardship Option (" StewardshipOption ")")  
    set Button2Monitor "Select Field"  
    set Button3Monitor "Apply Stewardship"  
    set Button4Monitor "Show Stewardship Areas"  
    set Button5Monitor "Define Crop Rotation"  
    set Button6Monitor "Unselect all Fields"  
    set Button7Monitor "Generate My Report!"  
    ]  
  
if Panel = "Maps and Settings"  
    ]  
    set Button1Monitor "Load Existing Map"  
    set Button2Monitor "Create Map from Scan or load GIS text file"  
    set Button3Monitor "Set Parameter Values"  
    set Button4Monitor "Load Setting"  
    set Button5Monitor "Save Setting"  
    set Button6Monitor "Show or hide Scale Bar"  
    set Button7Monitor "Initial 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.

```
]

```

```
if Panel = "Display Options"

```

```
[

```

```
  set Button1Monitor "Default view"
  set Button2Monitor "Show Nectar Visits"
  set Button3Monitor "Show Pollen Visits"
  set Button4Monitor "Identify!"
  set Button5Monitor "Show Input Files"
  set Button6Monitor "Show Parameter Values"
  set Button7Monitor "More Display Options"

```

```
]

```

```
if Panel = "Advanced Input Options"

```

```
[

```

```
  set Button1Monitor (word "Set Random Seed (" RAND SEED ")")
  set Button2Monitor "Advanced Setup Options"
  set Button3Monitor "Add a Background Image"
  set Button4Monitor ""
  set Button5Monitor ""
  set Button6Monitor ""
  set Button7Monitor "VERSION TEST"

```

```
]

```

```
if Panel = "My Own Maps"

```

```
[

```

```
  set Button1Monitor (word "Load Map 1 (" MySavedMap1 ")")
  set Button2Monitor (word "Load Map 2 (" MySavedMap2 ")")
  set Button3Monitor (word "Load Map 3 (" MySavedMap3 ")")
  set Button4Monitor (word "Load Map 4 (" MySavedMap4 ")")
  set Button5Monitor (word "Load Map 5 (" MySavedMap5 ")")
  set Button6Monitor (word "Load Map 6 (" MySavedMap6 ")")
  set Button7Monitor "Delete one of My Maps"

```

```
]

```

```
if Panel = ""

```

```
[

```

```
  set Button1Monitor ""
  set Button2Monitor ""
  set Button3Monitor ""
  set Button4Monitor ""
  set Button5Monitor ""
  set Button6Monitor ""
  set Button7Monitor ""

```

```
]

```

```
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 PanelButtonProc [buttonNumber]

```
let myButtonCommand ""
if buttonNumber = 1 [ set myButtonCommand Button1Monitor ]
if buttonNumber = 2 [ set myButtonCommand Button2Monitor ]
if buttonNumber = 3 [ set myButtonCommand Button3Monitor ]
if buttonNumber = 4 [ set myButtonCommand Button4Monitor ]
if buttonNumber = 5 [ set myButtonCommand Button5Monitor ]
if buttonNumber = 6 [ set myButtonCommand Button6Monitor ]
if buttonNumber = 7 [ set myButtonCommand Button7Monitor ]

;; ----- PANEL:
"STEWARDSHIP OPTIONS"

;; PANEL: "STEWARDSHIP OPTIONS"

if myButtonCommand = "Select Field" [ ButtonSelectFieldProc ]

if member? "Select Stewardship Option" myButtonCommand
  [
    set StewardshipOption user-one-of "Select a Stewardship Option" [ "legume" "margin" "plot" ]
    set Button1Monitor (word "Select Stewardship Option (" StewardshipOption ")")
  ]

if myButtonCommand = "Unselect all Fields"
  [
    foreach SelectedFieldsList [ ask foodsources with [ masterpatchID = ? ][ set shape "circle" ] ]
    set SelectedFieldsList []
  ]
if myButtonCommand = "Apply Stewardship" [ ButtonStewardshipOptionsProc ]
if myButtonCommand = "Show Stewardship Areas" [ StewardshipAreasProc ]
if myButtonCommand = "Generate My Report!" [ ButtonGenerateOutputProc ]
if myButtonCommand = "Define Crop Rotation" [ CropRotationSelectFilesProc ]

;; ----- PANEL: "MODIFY
MAPS"

;; PANEL: "MODIFY MAPS"
if myButtonCommand = "Clear whole Map"
  [
    ask turtles [ hide-turtle ]
    ask patches [ set pcolor grey ]
  ]
if member? "Choose Colour" myButtonCommand [ ButtonChooseColourProc ] ; use member?
command, as "Choose Colour" is only a part of the myButtonCommand
if myButtonCommand = "Draw on Map" [ DrawProc ]
if member? "Set Brush Size" myButtonCommand [ ButtonBrushSizeProc ]
if myButtonCommand = "Update current Map"
  [
    ;; ButtonDefineFarmareaProc currently not in use
    ButtonUseCurrentMapProc
  ]
```

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 myButtonCommand = "Switch Brush Shape to Circular"
_]
  set CircularBrush? true
  set Button3Monitor "Switch Brush Shape to Square"
  DrawProc
_]
if myButtonCommand = "Switch Brush Shape to Square"
_]
  set CircularBrush? false
  set Button3Monitor "Switch Brush Shape to Circular"
  DrawProc
_]
if myButtonCommand = "Replace Colours" [ ButtonReplaceColoursProc ]

;; ----- PANEL: "MAPS &
SETTINGS"

;; PANEL: "MAPS & SETTINGS"
if myButtonCommand = "Load Existing Map" [ ButtonLoadExistingMapProc ]
if myButtonCommand = "Save Setting" [ SaveLoadSettingsProc "Save!" "" ]
if myButtonCommand = "Load Setting"
_]
  SaveLoadSettingsProc "Load!" ""
  SaveLoadSettingsProc "Save!" ""
_]
if myButtonCommand = "Create Map from Scan or load GIS text file"
  [ BS ImportNewMapProc TRUE ] ; true: user input required!
if myButtonCommand = "Show or hide Scale Bar" [ ButtonScaleBarProc ]
if myButtonCommand = "Initial Queens" [ ButtonInitialQueensProc ]

;; ----- PANEL: "DISPLAY
OPTIONS"

;; PANEL: "DISPLAY OPTIONS"
if myButtonCommand = "Default view" [ ButtonDisplayProc "defaultView" ]
if myButtonCommand = "Show Nectar Visits" and max [cumulNectarVisits ] of foodsources > 0 [
ButtonDisplayProc "nectarVisits" ]
if myButtonCommand = "Show Pollen Visits" and max [cumulPollenVisits ] of foodsources > 0 [
ButtonDisplayProc "pollenVisits" ]
if myButtonCommand = "Identify!" [ ButtonIdentifyProc ]
if myButtonCommand = "Show Parameter Values"
_]
  foreach AllParametersList
_]
    let parameter remove " " ? ; some parameters have a blank added to the end of their name
(why?), which is removed here
    let command (word "set GenericRunCommandValue " parameter)
    run command
    output-print (word parameter ": " GenericRunCommandValue)
```


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 myButtonCommand = "Show Input Files"
[
  output-type "Input file: " output-print FoodsourcesFile
  output-type "Parameter file: " output-print MyParametersFile
  output-type "Habitat file: " output-print HabitatsFile
  output-type "Flower species file: " output-print FlowerspeciesFile
  output-type "Bumblebee species file: " output-print BeespeciesFile
  output-type "TextMap file: " output-print TextMap
]

if myButtonCommand = "More Display Options" [ ButtonDisplayButtonsProc ]

;; ----- PANEL:
"ADVANCED INPUT OPTIONS"

;; PANEL: "ADVANCED INPUT OPTIONS"
if myButtonCommand = "Advanced Setup Options" [ ButtonAdvancedSetupOptionsProc ]
if myButtonCommand = "Add a Background Image" [ ButtonBackgroundImageProc ]
if myButtonCommand = "Set Parameter Values" [ ParametersSetManuallyProc ]
if member? "Set Random Seed " myButtonCommand
[
  set RAND SEED read-from-string user-input "New value for RAND SEED (if 0: random-seed is not
set): "
  set Button1Monitor (word "Set Random Seed (" RAND SEED ")")
]
if myButtonCommand = "VERSION TEST" [ VersionTestProc ]
if myButtonCommand = "" [ ]

;; ----- PANEL: "MY
OWN MAPS"

;; PANEL: "MY OWN MAPS"
if myButtonCommand = "Delete one of My Maps"
[
  let deleteCommand user-one-of "Choose which of your saved maps should be deleted: " [ "Delete
My Saved Map 1" "Delete My Saved Map 2" "Delete My Saved Map 3" "Delete My Saved Map 4"
"Delete My Saved Map 5" "Delete My Saved Map 6" "Delete ALL My Saved Maps!" ]
  if deleteCommand = "Delete My Saved Map 1" [ set MySavedMap1 "" ]
  if deleteCommand = "Delete My Saved Map 2" [ set MySavedMap2 "" ]
  if deleteCommand = "Delete My Saved Map 3" [ set MySavedMap3 "" ]
  if deleteCommand = "Delete My Saved Map 4" [ set MySavedMap4 "" ]
  if deleteCommand = "Delete My Saved Map 5" [ set MySavedMap5 "" ]
  if deleteCommand = "Delete My Saved Map 6" [ set MySavedMap6 "" ]
  if deleteCommand = "Delete ALL My Saved Maps!"
[
  set MySavedMap1 ""
  set MySavedMap2 ""
```

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 MySavedMap3 ""
set MySavedMap4 ""
set MySavedMap5 ""
set MySavedMap6 ""
_]
PanelSettingProc
_]
if member? "Load Map 1" myButtonCommand
_]
ifelse MySavedMap1 = ""
[ set MySavedMap1 MyMap ]
[ set MyMap MySavedMap1 ]
Setup
_]
if member? "Load Map 2" myButtonCommand
_]
ifelse MySavedMap2 = ""
[ set MySavedMap2 MyMap ]
[ set MyMap MySavedMap2 ]
Setup
_]
if member? "Load Map 3" myButtonCommand
_]
ifelse MySavedMap3 = ""
[ set MySavedMap3 MyMap ]
[ set MyMap MySavedMap3 ]
Setup
_]
if member? "Load Map 4" myButtonCommand
_]
ifelse MySavedMap4 = ""
[ set MySavedMap4 MyMap ]
[ set MyMap MySavedMap4 ]
Setup
_]
if member? "Load Map 5" myButtonCommand
_]
ifelse MySavedMap5 = ""
[ set MySavedMap5 MyMap ]
[ set MyMap MySavedMap5 ]
Setup
_]
if member? "Load Map 6" myButtonCommand
_]
ifelse MySavedMap6 = ""
[ set MySavedMap6 MyMap ]
[ set MyMap MySavedMap6 ]
Setup
_]
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 GISexportDataProc [ data ] ;; currently not in use
;; ; corners of the (minimal) actual map:
;; let min-x min [ pxcor ] of patches with [ pcolorsave != 5 ]
;; let max-x max [ pxcor ] of patches with [ pcolorsave != 5 ]
;; let min-y min [ pycor ] of patches with [ pcolorsave != 5 ]
;; let max-y max [ pycor ] of patches with [ pcolorsave != 5 ]
;; let leftMargin (max-pxcor - GIS_ncols) / 2
;; let bottomMargin (max-pycor - GIS_nrows) / 2
;; if data = "visits"
;; [
;;   let dataList [ "Visits" ] ; "Nectar" "Pollen" ]
;;   foreach dataList
;;   [
;;     let filename (word " GISvisits " ? ".txt")
;;     let dataltem ? ; i.e. either "Nectar" or "Pollen"
;;     if file-exists? filename [ file-delete filename ]
;;     file-open filename
;;
;;     ; file-print ColourCodeList
;;     file-type "ncols" " file-print GIS_ncols
;;     file-type "nrows" " file-print GIS_nrows
;;     file-type "xllcorner" " file-print GIS_xllcorner
;;     file-type "yllcorner" " file-print GIS_yllcorner
;;     file-type "cellsize" " file-print BS_Scaling_m/NLpatches
;;     file-type "NODATA_value" " file-print GIS_NoDataValue
;;
;;     let xpos 0 ; keeps track of number of entries in each line of outfile
;;     foreach sort patches
;;     [
;;       ask ?
;;       [
;;         ; if (pxcor = leftMargin - 1) and (pycor >= bottomMargin and pycor < GIS_nrows +
bottomMargin)
;;         ; [ file-type (word GIS_NoDataValue " ") ]
;;         if (pxcor >= leftMargin and pxcor < GIS_ncols + leftMargin) and (pycor >= bottomMargin and
pycor < GIS_nrows + bottomMargin)
;;         [
;;           if xpos = 0 [ file-type (word GIS_NoDataValue " ") ] ; first (and last) column needs to be -9999
;;           let memoID flowerpatchID
;;           ifelse (flowerpatchID = -1 or count foodsources with [ who = flowerpatchID ] = 0)
;;           [ file-type -1 ]
;;           [
;;             let cumulVisitsTotal sum [ cumulNectarVisits ] of foodsources with [ masterpatchID =
memoID ]
;;             if dataltem = "Pollen"
```

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 cumulVisitsTotal sum [ cumulPollenVisits ] of foodsources with [ masterpatchID =
memoID ] ]
;; if dataItem = "Visits"
;; [ set cumulVisitsTotal sum [ cumulPollenVisits + cumulNectarVisits ] of foodsources with [
masterpatchID = memoID ] ]
;; let cumulVisitsTotalPerGridcell 0
;; if any? foodsources with [ who = memoID ]
;; [ set cumulVisitsTotalPerGridcell (cumulVisitsTotal / [ area sqm ] of foodsource
flowerpatchID) ; visits per m2
;; * (1 / Scaling_NLpatches/m) ^ 2 ] ; -> visits per grid cell
;; file-type precision cumulVisitsTotalPerGridcell 1
;; ]
;; file-type " "
;; set xpos xpos + 1
;; if xpos >= GIS_ncols ; + leftMargin ;; max-pxcor ; start a new line
;; [
;; set xpos 0
;; file-print GIS_NoDataValue ; last column: -9999
;; ; if pycor != GIS_nrows + bottomMargin [ file-type (word GIS_NoDataValue " ") ] ; first
column: -9999
;; ]
;; ]
;; ] ; ask ?
;; ]
;; file-close
;; ]
;; ]
;;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 ]
ask bees[
set knownMasterpatchesNectarList []
set knownMasterpatchesPollenList []
set nectarsourceToGoTo -1
set pollenSourceToGoTo -1
set currentFoodsource -1

]
ask colonies
[
set masterpatchesInRangeList []

```

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 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 ListFromStringREP [inputList]

```
; returns a string into a list (with blanks as separator)
let resultList []
let remainingInputString inputList
while [ member? " " remainingInputString = true ]
[
let cutHere position " " remainingInputString
ifelse cutHere > 0 ; in case remainingRotationsString starts with a blank
]
set resultList lput substring remainingInputString 0 cutHere resultList
set remainingInputString substring remainingInputString (cutHere + 1) length
remainingInputString
]
[ set remainingInputString but-first remainingInputString ]
]
set resultList lput remainingInputString resultList
report resultList
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 CropRotationSelectFilesProc

let nRotations read-from-string user-input "Insert number of ADDITIONAL foodsource files you would you like to use (insert 0 to remove rotations): "

let rot 1

set CropRotationList []

if nRotations > 0 [set CropRotationList lput FoodsourcesFile CropRotationList]

repeat nRotations

[

user-message "Choose a ' Foodsources' text file (this step might be repeated)"

let nextFile FilenameREP true

set CropRotationList lput nextFile CropRotationList

]

set CropRotationListAsString ""

foreach CropRotationList [set CropRotationListAsString (word CropRotationListAsString " " ?)]

if user-yes-or-no? "Save settings now?" [SaveLoadSettingsProc "Save!" ""]

end

, *****

to ParametersSetManuallyProc

; called by button

foreach AllParametersList

[

let parameter remove " " ? ; some parameters have a blank added to the end of their name (why?), which is removed here

let command (word "set GenericRunCommandValue " parameter)

run command

output-print (word parameter ": " GenericRunCommandValue)

]

let parameterType "undefined"

let parameterToBeChanged user-one-of "Select parameter you would like to change (press 'Setup' when finished): " ; NOTE: only a selection of parameters are shown here

[

"AbundanceBoost"

"BeeSpeciesInitialQueensListAsString" ; e.g. "Bee_longTongue 500 Bee_shortTongue 500"

BEESPECIES

"COLONIES IBM"

"CumulVisitsOnlyLastYear?"

"FlowerspeciesFile"

"FoodSourceLimit"

"ForagingMortalityFactor"

"ForagingMortalityModel"

"Gridsize"

"HabitatsFile"

"KeepDeadColonies?"

"Lambda detectProb"

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*.

```
"MapAreaIncluded"
"MasterSizeFactor"
"MaxForagingRange_m"
"MaxHibernatingQueens"
"MaxPatchRadius_m"
"MergeHedges?"
"MinSizeFoodSources?"
"N_Badgers"
"N_Psithyrus"
"RemoveEmptyFoodSources?"
"SexLocus?"
"ShowCohorts?"
"ShowDeadCols?"
"ShowFoodsources?"
"ShowGrid?"
"ShowInspectedColony?"
"ShowMasterpatchesOnly?"
"ShowNests?"
"ShowPlots?"
"ShowQueens?"
"ShowSearchingQueens?"
"ShowWeather?"
"BeespeciesFile"
"StopExtinct?"
"UnlimitedMales?"
"Weather"
_]
let newValue user-input "Set new value for this parameter (leave blank to cancel)"
if newValue != "" [ run (word "set " parameterToBeChanged " " newValue) ]
SaveLoadSettingsProc "Save!" ""
end
```

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

;; =====
;; END OF THE CODE   +++   END OF THE CODE   +++   END OF THE CODE   +++
;; =====
```

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