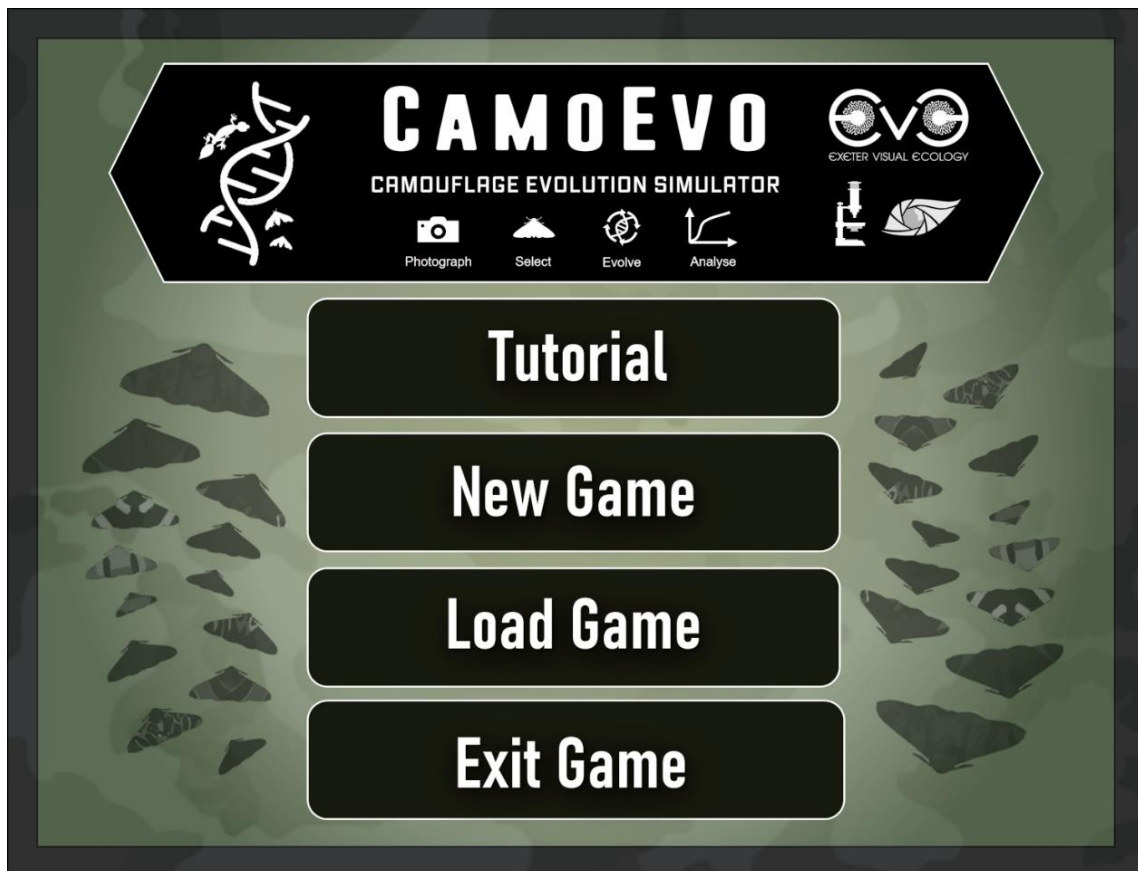


CamoEvo Version 2.0



Contents

Changes Summary	3
CamoEvo	3
Fixes.....	3
Additions.....	3
ImageGA.....	3
Additions.....	3
Psychophysics Game	3
Additions.....	3
Image Analysis	4
Additions.....	4
Pattern Generation	4
Fixes.....	4
Additions.....	4
Camo Print	4
Evolution Module.....	4
Updated Plots	5
LAB Measures.....	5
Phenotype Distribution.....	5
Gene Ranges.....	6
Demo (Peppered Moth)	6
Evolution Module	7
Random Selection Demo.....	7
Table Entry Selection Demo	8
Spawn Maps	8
Spawn Methods	9
Measure	9
Grid.....	10
CamoPrint.....	11
Pattern Generators.....	15
Animals 2	15
Eggs 2.....	15

Changes Summary

CamoEvo

Fixes

- **Analysis Plots**, revised the plot formats to be clearer.
- **GabRat** now works with square shaped targets for image analysis, note (does not fix the issue for RAW GabRat).
- **ColourRange** no longer produces NaNs if the range=0.
- **TriangleShape** now uses the correct setting.
- **Distribution** fixed the default gene distribution to be uniform.
- **Built in Games** made it so the built in games have all the Population_Settings
- **GA interface** fixed error with GA best clone entry
- **Rank** CamoEvo no longer defaults the fitness value to rank

Additions

- **Updated UI**, new cleaner UI with more options when completing a game (show phenotype plot/gif) and a new phenotype distribution plot for the L,a,b and contrast (L stdv) metrics.
- **Repeats**, CamoEvo can now function with repeats for each target.
- **PepMoth**, new starting default showcasing the new transform features and the Gaussian starting population.
- **Shapes**, added a new egg clutch and a square shape to the defaults.
- **Transform Phenotypic Ranges** similar to specifying the colour ranges you can now reduce the phenotypic ranges of genes. E.g. making it so 0.0-1.0, is instead expressed as 0.2-0.4 or 0.5-1.0 etc. To make the transform.txt files copy the gene template and replace the second row with the *Min* values and the third row with the *Max* values, leaving the title unaltered. "Transform_PhenotypeRanges.txt"

ImageGA

Additions

- **Average Repeats**, if there are multiples of the same target ID, ImageGA will average the fitness values
- **Normal Distribution** new normal distribution starting population 'NormalDist', useful for evolution practicals.
- **Relabelling**, the rescue system no longer renames the target unless clonal mutation is enabled.
- **Transform Starting Gene Ranges** similar to specifying the colour ranges you can now specify the starting ranges for all of the genes. To make the transform.txt files copy the gene template and replace the second row with the *Min* values and the third row with the *Max* values, leaving the title unaltered. "Transform_StartGeneRanges.txt"

Psychophysics Game

Additions

- **Improved Spawning**, targets now spawn using exclusion-based system, preventing targets from overlapping
- **Mouse Click**, can no longer just hold the mouse down.
- **N misses**, can now set the number of misses (miss.N), where N = number of misses (1 = 1 click)
- **Spawn Maps**, allows the user to assign regions of an image that targets can spawn and the angle that they should spawn in.
- **Spawn Systems**, new grid, colour distance and proximity based spawn systems.
- **New Image Modifiers**, drop shadow and pre-game modifiers (alters the saved image).

Image Analysis

Additions

- **Batch Measure** now separated into measurement macros. These are run by using the *0 Batch Measure Population* Macro. First select a population, then select a measure from the table. The analysis in the UI now uses the LAB measure and imports the data to reconstruct the plots. All data is outputted in a Data_Measures_[PopName].txt file.

*If you want to run additional analyses you will need to **rename** or **remove**. The existing files.*

- **Image Analysis Folder.** The image analysis scripts have been moved to a new folder in 2 Camo Req.
- **Measure Macros** New measurement scrips.
 - o *Revised LAB measure*
 - o *New XYZ measure, the same as the LAB measure but with the XYZ space*
 - o *New RNL local measure, the RNL chromaticity similarity.*

Pattern Generation

Fixes

- **Animal Patterns** fixed issue with incorrect thresholding of patterns.

Additions

- **Gene Ranges.** Altered existing scripts to allow for starting gene ranges.
- **Experimental Egg V2.** A slower but more effective egg pattern generator
 - o *Both maculations have all the same genes, median and random walk*
 - o *Added new background speckling*
 - o *Improved the pore texture*
 - o *Introduced a wave form to the colour space, causing saturation to peak then drop with deposition. (Creates more natural looking eggs).*
 - o *New clutch feature, if the target shape has multiple objects, it'll instead produce a radial pattern.*
- **Experimental Animal V2.** A slower but more dynamic animal pattern generator
 - o *Both maculations and background colours can be gradiented.*
 - o *Maculation gradients are now mediated by reaction-diffusion not a circular mask.*
 - o *Added sin wave distortion on the x and y axis*
 - o *Edge enhancement can now be offset on the x and y ais.*
 - o *Size of the reaction-diffusion map is increased to 4000px*
 - o *Scale of the target is now flexible.*

Camo Print

Print Sheets, new system designed for the purposes of creating printed prey.

Evolution Module

Evolve Loop, designed to allow for looping pattern generation as if running the CamoEvo Game

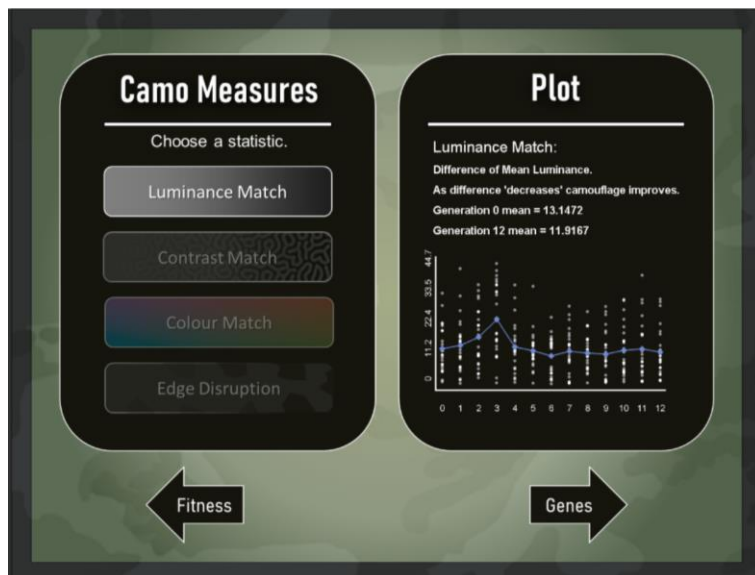
Calculators, custom fitness calculators e.g. data entry and random fitness. For driving evolution without the psychophysics game.

Updated Plots

LAB Measures

The plotting system for CamoEvo has been updated to instead use a scatter plot with an average trend line (blue).

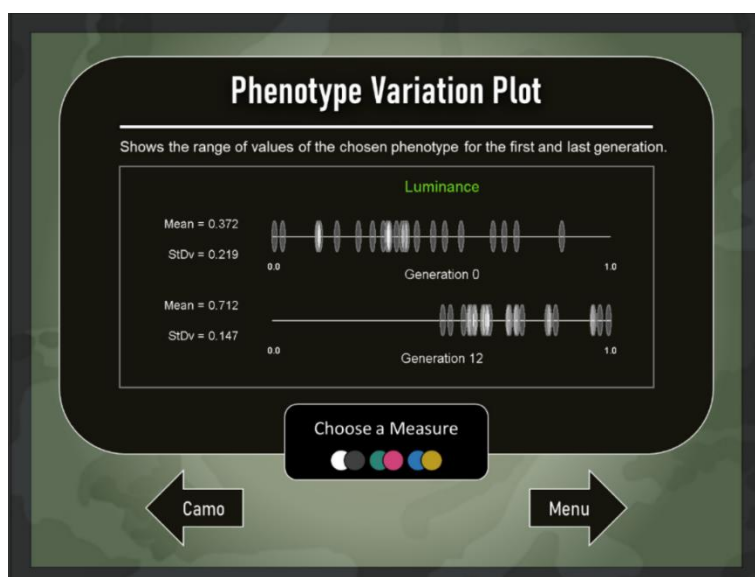
Figure, example luminance plot showing 12 generations.



Phenotype Distribution

You can now plot the normalised distribution of luminance, green-red, blue-yellow and luminance contrast for the targets in a similar fashion to the gene plots. The purpose of this is to provide a simplified demonstration of the direction of selection for practicals.

Figure, the above shows the mean target luminance in the first generation and below the mean luminance in the last generation.



Gene Ranges

Demo (Peppered Moth)

To provide an easier mechanism of modifying the appearance space of the targets without having to create a custom phenotype make, a gene range editor has been added.

Users of CamoEvo can now manipulate the starting gene ranges (ranges of the starting population) and the gene exploration (explorable space for future generations) ranges.

To use this feature you will need to create a copy of the gene template for the phenotype maker and then rename it **Transform_StartGeneRanges.txt** for the starting gene ranges and **Transform_PhenotypeRanges.txt** for the exploration space.

e.g. Template Animals

ID	P1	P2	Sex	ptn_dim_xcp	ptn_dim_ycp	ptn_dim_wdt	ptn_dim_asr	ptn_dim_agl
-	-	-	-	0	0	0	0	0

Then replace sex with Range, the first row gives the minimum value and the second row gives the maximum value. For the usual range use min=0 and max =1

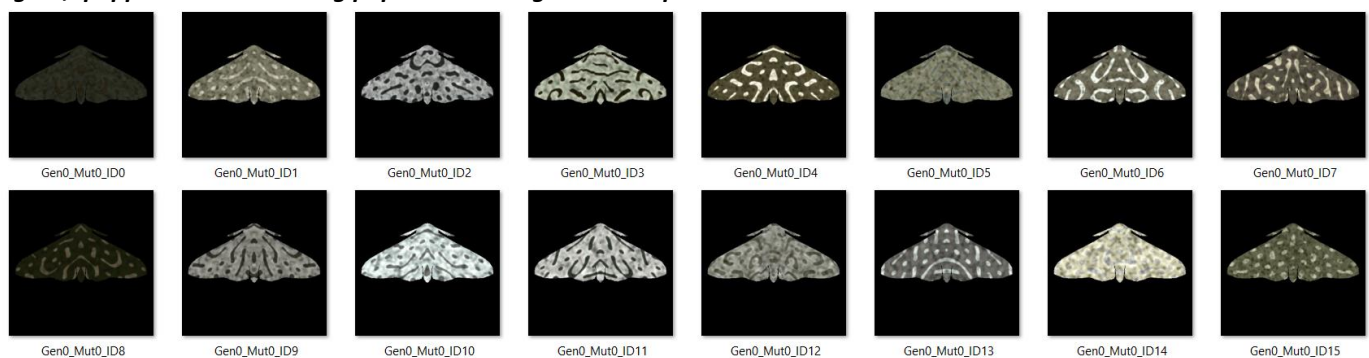
e.g. Transform_

-	-	-	Range	ptn_dim_xcp	ptn_dim_ycp	ptn_dim_wdt	ptn_dim_asr	ptn_dim_agl
-	-	-	Min	0.5964	0.37	0.67	0.57	0.6154
-	-	-	Max	0.6564	0.43	0.73	0.63	0.6754

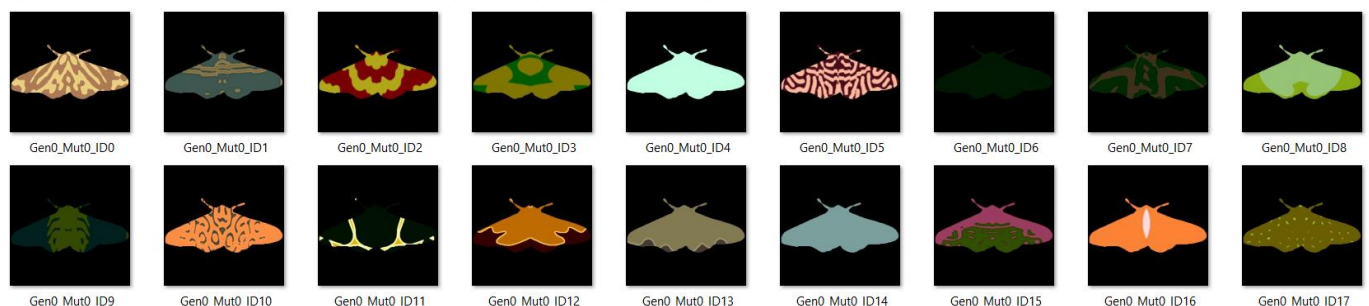
When using StartGeneRanges, the Gen_0 starting population values will be shifted to the new range but keeping their original distribution. When using the PhenotypeRanges the values won't change but the function of the value will. E.g. in a range of 0-0.5, a gene equalling 1 will in function equal 0.5.

For determining which genes to turn off use the gene reference table in the CamoEvo handbook and the phenotype maker plugin. To test whether you are happy with the ranges I recommend

Figure, 'peppered moth starting population' using restricted pattern dimensions.



Figure, custom moths rendered with the speckling and gradient genes turned off.

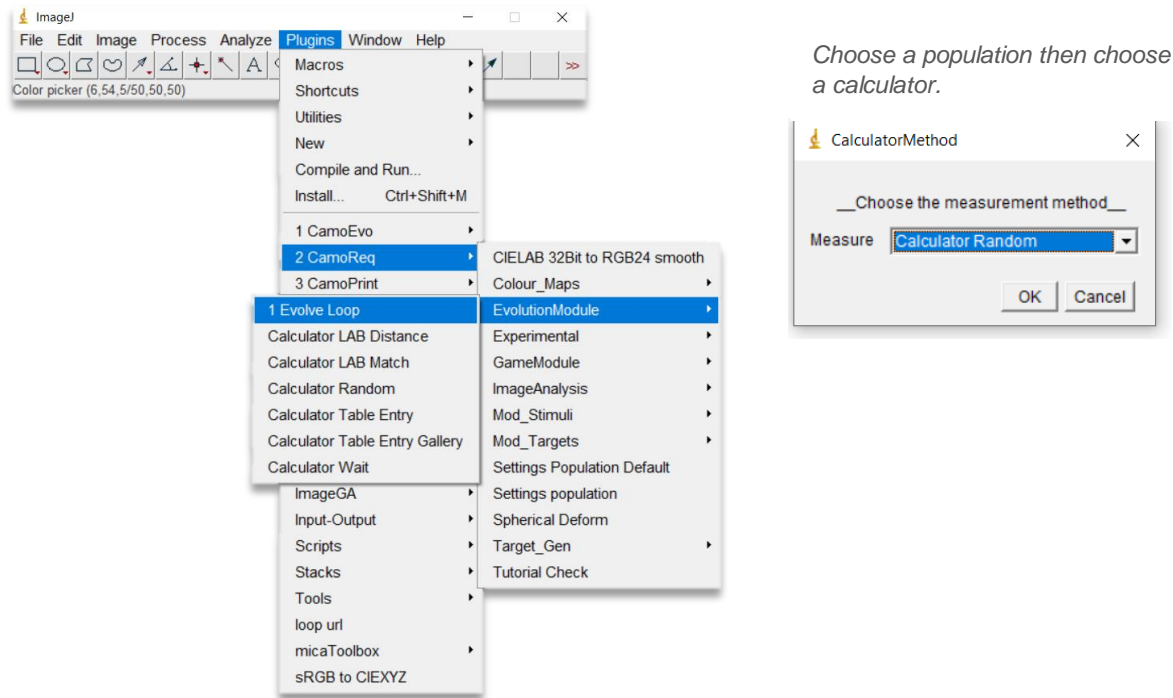


Evolution Module

2 CamoReq/ EvolutionModule/ 1 Evolve_Loop

Similar to how the psychophysics game can be used independently of CamoEvo, the pattern evolution loop can now be run independently of the psychophysics game. This feature requires a population folder, with the population settings as per usual. But rather than rendering the stimuli on backgrounds it will instead run one of the selected fitness calculators.

To make your own fitness calculator create a plugin titled '*Calculator_[Insert Name].txt*' and place it in the 2 CamoReq using one of the existing calculators as a base.



Random Selection Demo

A common test in pattern evolution experiments is to generate prey under random selection to prove that improvements are due to selection and not drift. The Calculator Random inputs random decimal values from 0-1.

Figure, 'peppered moth starting population' evolved for 10 generations using the random calculator

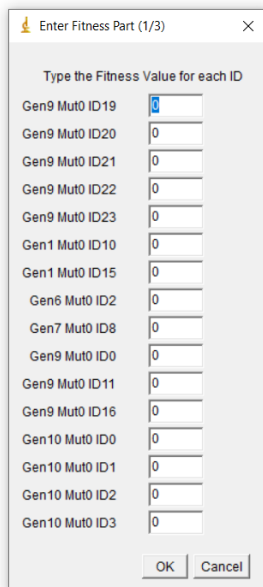


Figure, 'peppered moth starting population' evolved for 10 generations using the game to select for camouflage



Table Entry Selection Demo

Allows the user to input their own fitness values using a table in ImageJ. If the table is too long it will split into multiple pages. Higher values = higher fitness. There are two versions one which just produces a table and 'Gallery' which uses the Phenotype Plotter to produce a table showing all the phenotypes in the most recent generation.

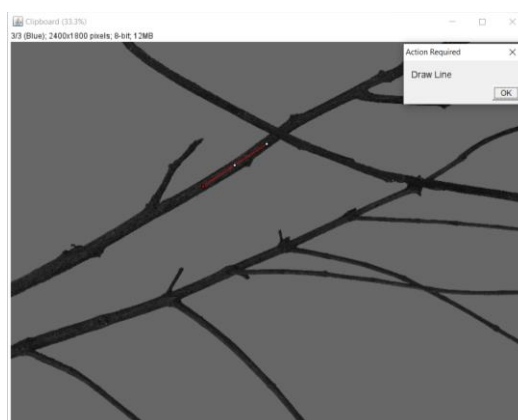
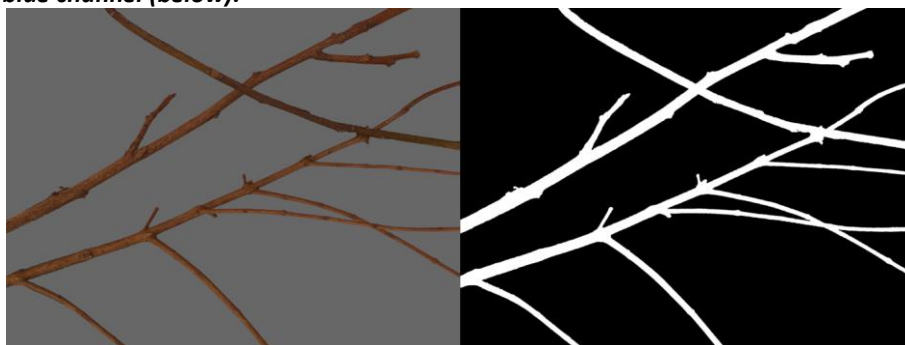


Type the Fitness Value for each ID	
Gen9 Mut0 ID19	<input type="text" value="0"/>
Gen9 Mut0 ID20	<input type="text" value="0"/>
Gen9 Mut0 ID21	<input type="text" value="0"/>
Gen9 Mut0 ID22	<input type="text" value="0"/>
Gen9 Mut0 ID23	<input type="text" value="0"/>
Gen1 Mut0 ID10	<input type="text" value="0"/>
Gen1 Mut0 ID15	<input type="text" value="0"/>
Gen6 Mut0 ID2	<input type="text" value="0"/>
Gen7 Mut0 ID8	<input type="text" value="0"/>
Gen9 Mut0 ID0	<input type="text" value="0"/>
Gen9 Mut0 ID11	<input type="text" value="0"/>
Gen9 Mut0 ID16	<input type="text" value="0"/>
Gen10 Mut0 ID0	<input type="text" value="0"/>
Gen10 Mut0 ID1	<input type="text" value="0"/>
Gen10 Mut0 ID2	<input type="text" value="0"/>
Gen10 Mut0 ID3	<input type="text" value="0"/>

Spawn Maps

You can now make spawn maps for your backgrounds. Spawn Maps are .png images that communicate where targets are allowed to go. To use spawn maps. Add a new folder to your population folder titled/ SpawnMaps/ then for each image create .png of the same size and scale. The red channel marks all the locations that targets can spawn. The green channel marks the angle that the target should spawn when using the orientatation spawn system. Spawn maps can easily be made using thresholding or manually drawing lines. To include angle open the original .png in imageJ and run the 2 CamoReq/Draw Angle plugin. This allows lines to be drawn with the angle value being automatically assigned to the red channel. Then convert back to a RGB image 'run("RGB Color")'.

Figure, example background (left) and spawn map (right). Targets can only spawn in the regions labelled in white. For angles you need to label the blue channel (below).



Spawn Methods

SettingsGame.txt

SpawnMethod =

Measure

As this spawn system requires LAB conversion it runs slower. This spawn system does not overlap spawn maps.

measure.L.near: targets choose the locations that are closer in luminance to the target

measure.A.near: targets choose the locations that are closer in green-red to the target

measure.B.near: targets choose the locations that are closer in blue-yellow to the target

measure.LAB.near: targets choose the locations that are closer in LAB to the target

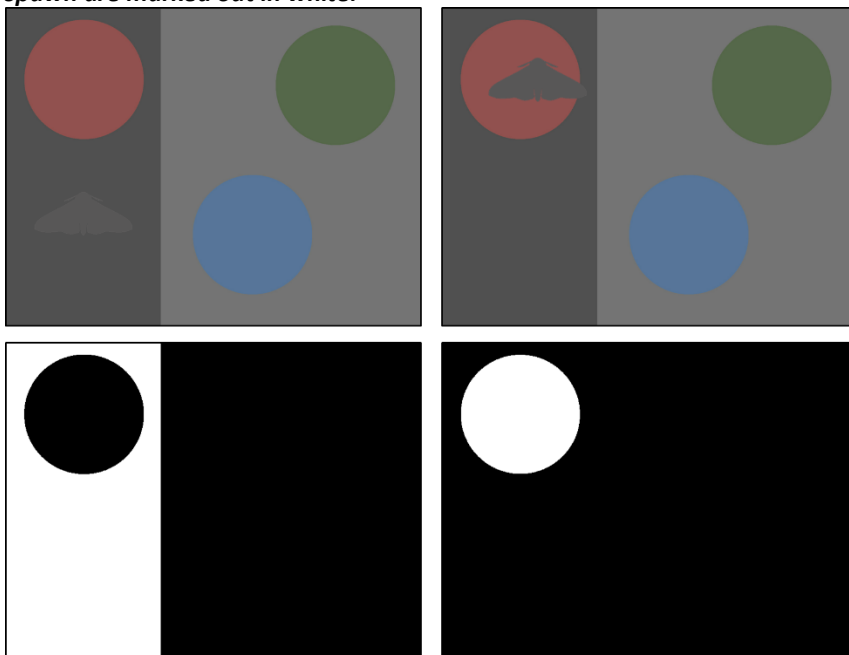
measure.L.far: targets choose the locations that are closer in luminance to the target

measure.A.far: targets choose the locations that are closer in green-red to the target

measure.B.far: targets choose the locations that are closer in blue-yellow to the target

measure.LAB.far: targets choose the locations that are closer in LAB to the target

Figure, example thresholds for spawn locations when set to LAB.near (left) and LAB.far (right). Areas where the grey moth will spawn are marked out in white.



Grid

grid.x.y: e.g. grid.2.2 targets spawn in a two by two grid in the centre of the creen.

Spawns targets that share the same slide in a choice grid against the backgrounds (up to six targets). To make this system work for evolution the following settings should be used.

Settings Game:

Adjust the grid to the number of targets.

Set the angle range to 0 so the targets all face the same direction.

Set the fail trigger miss.N, where N is the number of targets that should survive in each choice grid.

Settings Population:

Invert the fitness value

Use rank as the fitness value

AlgorithmSettings:

Make the number that survive equal to the number chosen. E.g. for a binary choice experiment with 24 targets, 12 should survive and 12 should die.

Figure, example three way choice test between three moth patterns. The one clicked on gets to survive.



Orientation

orientation.positive aligns to the background

orientation.positive aligns perpendicular to the background

In nature animals frequently align themselves to the orientation of surrounding objects. By setting the spawn system to orientation.positive targets will align to their background by choosing an orientation that best matches the mean of what behind them.

If you use a spawn map that features angles, then the targets will orientate using the angles provided.

CamoPrint

3 CamoPrint/

Is a new set of plugins that can be used to help create printed prey.

- 0)_Regenerate_Prey_Patterns: *Recreates the prey patterns for the most recent generation.*
- 0)_Regenerate_Prey_Sheets: *Recreates the prey sheet for the most recent generation.*
- 1)_CamoPrint_Setup: *Sets up the folder directories and populations for CamoPrint*
 - Select Number of Treatments, Repeats and whether they should share starting populations.
 - Choose whether to group targets into subdivisions.
 - Adjust the appearance of the printer sheets and how to label the targets.
- 2)_2)_Population_Fitness_Submission: *Creates a table for the user to enter the fitness values for the current gen*

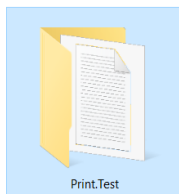
Paper Caterpillars Demo

This is not a recommendation for experimental design, simply an example of the print features.

In this demo you will learn how to create print sheets using the CamoPrint plugins. The premise of the experiment is simple. A set of six host plants have been setup in the field, each plant will have 4 caterpillars on them. The plants are monitored continuously with cameras and checked daily. Once two caterpillars have been predated per plant their ranking are submitted the first two predated get a rank of 0 and those that are last get a rank of 1.

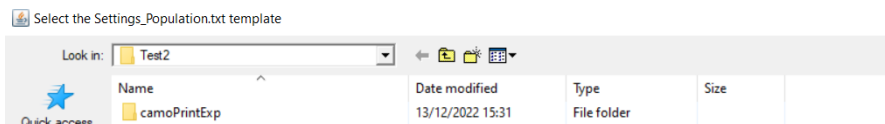
Step 1. Create a Directory

This will be where CamoPrint is stored.

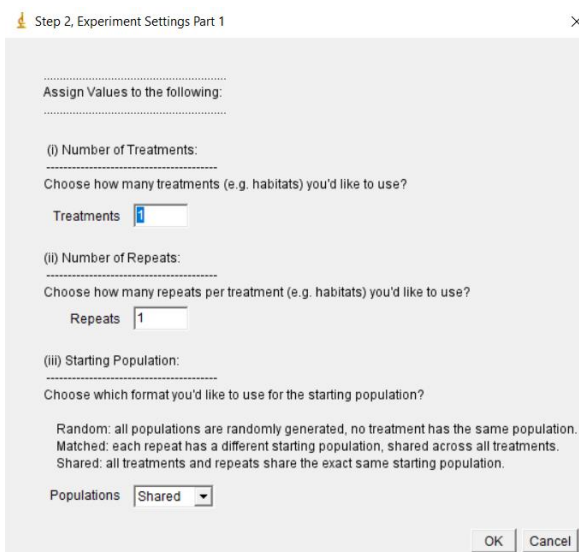


Step 2. Run 1)_CamoPrint_Setup

Select a Population Settings template (same settings as used in CamoEvo)



You can now choose how many treatments there are and repeats (1&1 in this case). Starting populations can be random, matched to repeat or shared across all treatments & repeats.



You can now choose how many targets should be in each group, (4 in this case) and how they should be grouped. Disassortative isn't currently operational.

Step 2, Experiment Settings Part 2

(iv) Group Size:
How many individuals are shown together, if individuals are all independent put 1.

Group Size

(v) Grouping Method:
How are individuals assigned to groups?

Random: individuals are randomly assigned.
Ranked: each group has an even distribution of individual fitnesses.
Disassortative: individuals are grouped disassortatively by genotype.

Grouping

OK Cancel

CamoPrint makes sheets that are automatically organised and tagged to make printing and grouping easier. You can adjust how many targets should be shown per row and how the rows should be space.

Step 2, Experiment Settings Part 3

(vi) Sheet Layout:
how do you want the printer sheet arranged?

Targets Per Row:

















Row Spacing (px):

Column Spacing (px):

Copies Per Sheet:

OK Cancel

Print_Habitat1_Repeat1

			
H1.1.1.R	H1.1.1.G	H1.1.1.B	H1.1.1.Y
			
H1.1.2.R	H1.1.2.G	H1.1.2.B	H1.1.2.Y
			
H1.1.3.R	H1.1.3.G	H1.1.3.B	H1.1.3.Y
			
H1.1.4.R	H1.1.4.G	H1.1.4.B	H1.1.4.Y

Example print sheet vertical black bars mark the end of a group.

If your treatment is divided up into grids you can have CamoPrint create randomised spawn locations. For example let's say each plant is divided up into a 10 by 10 square and the targets should not spawn too close to one another (can't be within two squares). If left-right is used then targets will be put in order from the left-right of the grid. Individuals within each group can be labelled by colour, letter or number.

Step 2, Experiment Settings Part 4

(vii) Spawn Layout:
how do you want the spawn grid arranged?

N rows (x):

N column (y):

Min Distance:

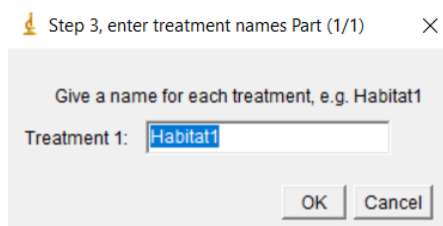
Spawn Method: (random, left-right)

(viii) Label Method:
how do you want the targets labelled

Label: (colour,alphabet,number)

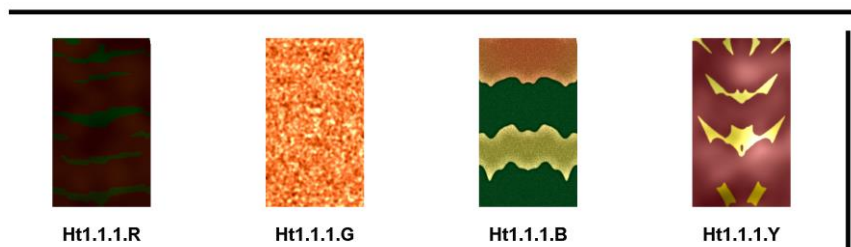
OK Cancel

Treatments can be named. Folders storing each treatment and repeat will be automatically named. E.g. Treat1_Repeat1, Treat1_Repeat2.



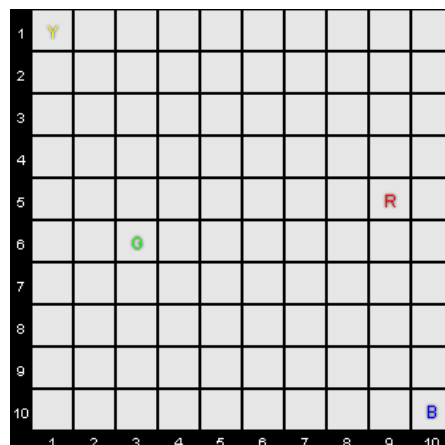
Target names are encoded as follows Abbreviated Treatment . Replicate Number . Group Number . Label (colour,letter,number)

Print_Habitat1_Repeat1

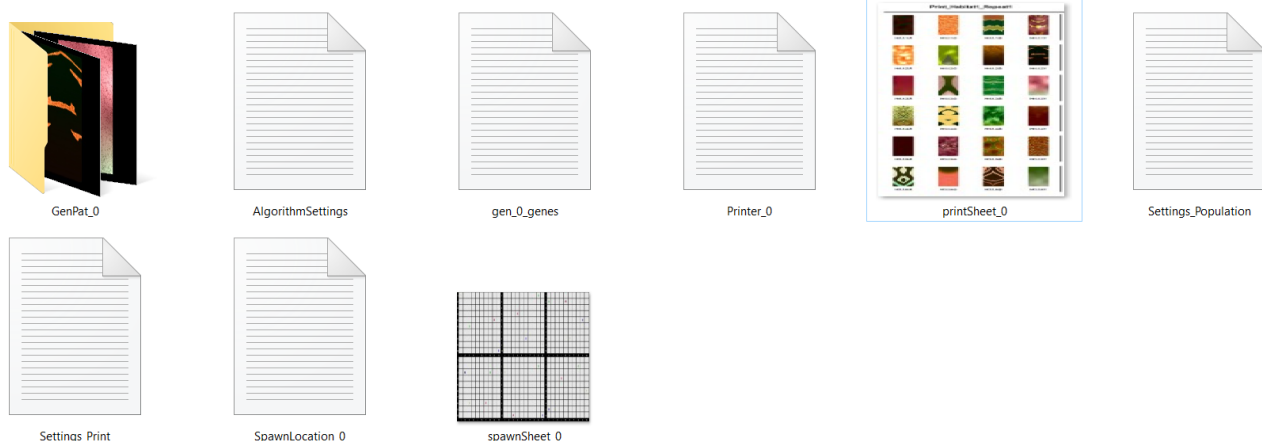


Once your setting shave been chosen for each group a spawn map will be made. As both a .png gallery and a (SpawnLocation).txt file. The names are encode in the .txt file as assigned (colour, letter, number), x coordinate, y coordinate.

Group: 1	R.9.5	G.3.6	B.10.10	Y.1.1
Group: 2	R.4.4	G.9.1	B.6.8	Y.10.9
Group: 3	R.5.2	G.1.2	B.9.5	Y.6.8
Group: 4	R.7.8	G.8.3	B.2.3	Y.3.8
Group: 5	R.3.10	G.9.2	B.10.10	Y.1.3
Group: 6	R.4.4	G.8.2	B.1.9	Y.10.6



Example file structure from Generation 0



Step 3. Create

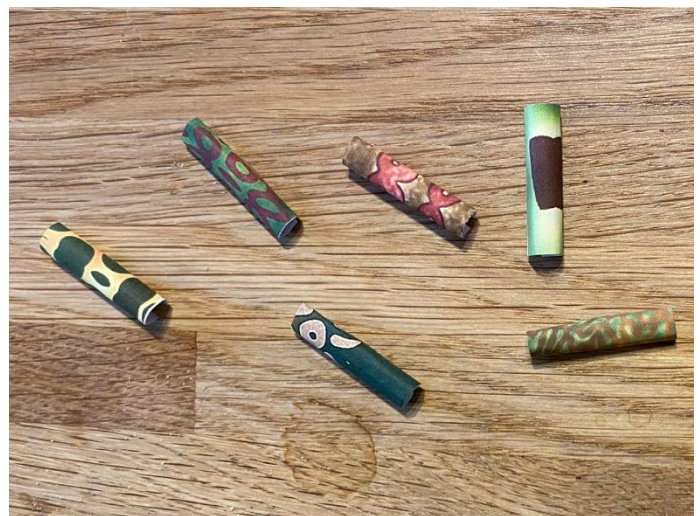
Algorithm settings will need to be manually such that the number deleted and breeding is in proportion to each group.

Turn off adaptive mutation based on fitness (fitness can't change)

Adjust the rescue number based on group size.

Printed targets should be made using the same printer and cartridges. Colour calibrated printing is not necessary as targets will evolve to the printers own colour space and colour should be measured with the printed stimuli, not the sRGB printer patterns.

Figure, example printed and rolled caterpillars



Step 4. 2) Population Fitness Submission

Once the fitness values of a generation select the treatment+replicate folder with the Fitness Submission plugin and enter the values for each target. Targets will be names with their assigned name and ImageGA name.

You can then enter the fitness values for each individual in group. Once complete, the next generation will be created automatically along with the spawn maps.

🔧 Step 2, Enter Fitness Part (1/3) ✕

Print_Habitat1_Repeat1

Type the Fitness Value for each ID

Group1

Ht1.1.1.R...Gen0 Mut0 ID14

Ht1.1.1.G...Gen1 Mut0 ID3

Ht1.1.1.B...Gen1 Mut0 ID1

Ht1.1.1.Y...Gen0 Mut0 ID16













Group2

Ht1.1.2.R...Gen1 Mut0 ID8

Ht1.1.2.G...Gen1 Mut0 ID4

Ht1.1.2.B...Gen1 Mut0 ID2

Ht1.1.2.Y...Gen1 Mut0 ID0

			
Ht1.1.1.R	Ht1.1.1.G	Ht1.1.1.B	Ht1.1.1.Y
			
Ht1.1.2.R	Ht1.1.2.G	Ht1.1.2.B	Ht1.1.2.Y
			
Ht1.1.3.R	Ht1.1.3.G	Ht1.1.3.B	Ht1.1.3.Y

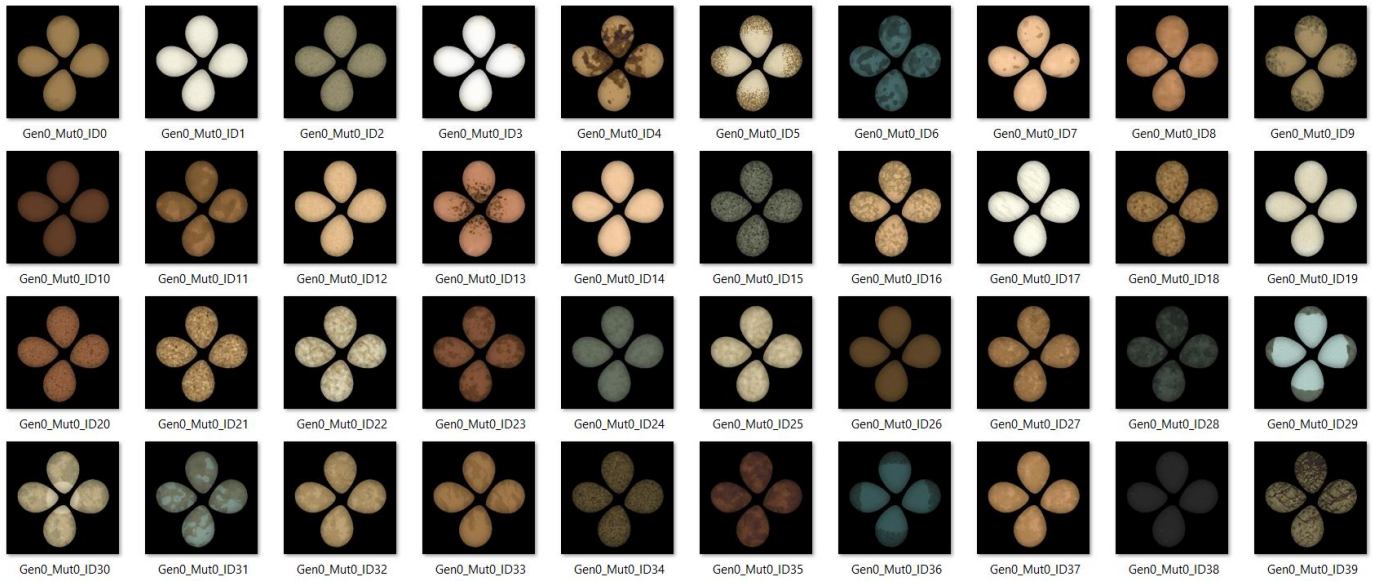
Pattern Generators

Eggs 2 (Experimental)

Features

Allows you to render clutches of eggs and applies the random walk genes to both maculations and uses an improved colour space where saturation drops off with luminance after a peak.

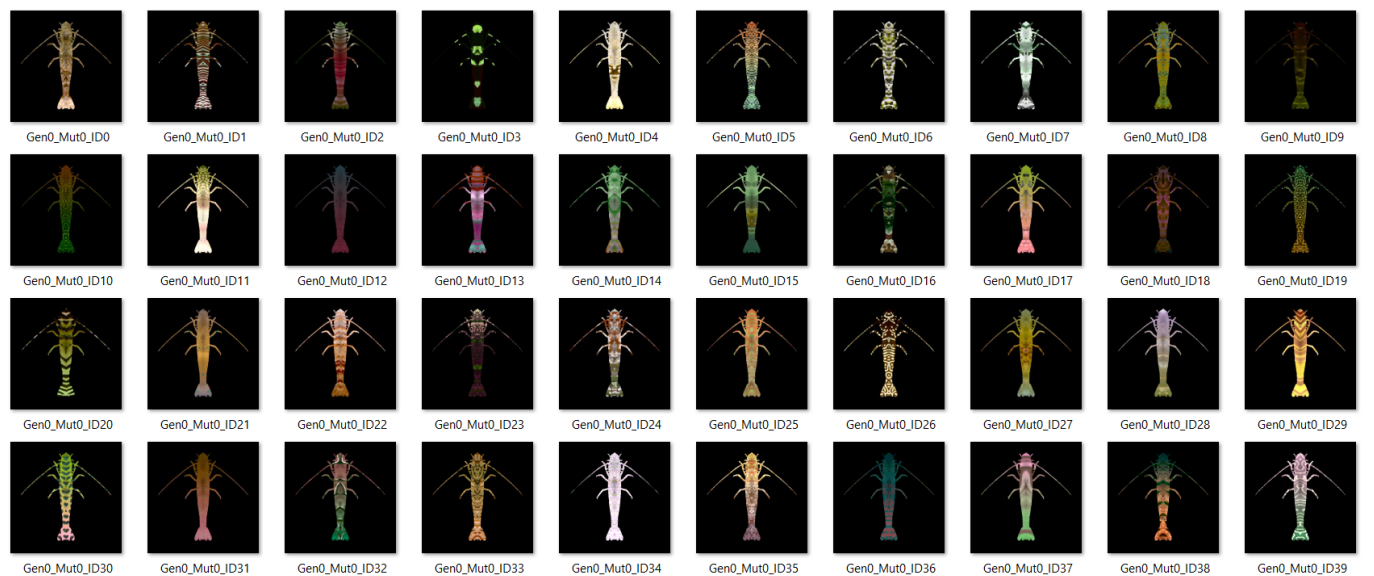
Figure, example clutches



Animals 2 (Experimental)

New system uses reaction-diffusion of reaction-diffusion to define pattern distribution, both maculation and background have two colours and edge enhancement is now defined by edge detection.

Figure, example prawns



Figure, example lizards



