Butterfly Mimicry

Kui.Chen 2016/02/15 Monday

Implementation

Options

These options are used to set the attributes and output format in Butterfly Camouflage Evolution.

```
In[1]:=
(*These 3 options are used to control the output format*)
Options[displayWorldSpot] = {
   BackgroundHue → 0.9, ColoringFunction → GrayLevel
  };
Options[butterflyGraphics] = {
   BackgroundHue → 0.9, ColoringFunction → GrayLevel
  };
Options[displayButterflyWorld] = {DisplayMatrixWidth → 0};


(*These options are used to control the properties in the evolution*)
Options[mutation] = {
   MutationProbability → 0.0, MutationRange → 0.1
  };
Options[reduction] := {BackgroundHue → 0.5};
Options[evolveButterflies] = {
   InitialPopulation → {},
   WorldWidth → 10,
   ButterflyProbability → 0.5,
   ReductionCreationPortion → 0.1,
   BackgroundHue → 0.5,
   ColoringFunction → Hue,
   DisplayMatrixWidth → 0,
   Generations → 10,
   MutationProbability → .2,
   MutationRange → 0.1
  };
```

Generating an Initial Butterfly Vector

Generating butterfly or tree trunk according to the random real num-

ber.{1, _} represents the butterfly, while the {0, BGColor} represents
the tree trunk.

In[7]:=
```
createWorld[width_: 100, butProb_: 0.5, opts___] := Table[
    If[RandomReal[] < butProb,
     {1, RandomReal[]},
     {0, BGColor}
     ],
    width
   ];
```

Test

Test createWorld function.

In[8]:=
```
butterflies = createWorld[10, 0.5]
```

Out[8]=
```
{{0, BGColor}, {0, BGColor}, {1, 0.570245}, {1, 0.360054}, {1, 0.884588},
  {1, 0.987648}, {0, BGColor}, {1, 0.171051}, {1, 0.189441}, {1, 0.301695}}
```

In[9]:=
```
butterflies // TableForm
```

Out[9]//TableForm=
```
0    BGColor
0    BGColor
1    0.570245
1    0.360054
1    0.884588
1    0.987648
0    BGColor
1    0.171051
1    0.189441
1    0.301695
```

Visualization

Display one butterfly on the tree trunk.

In[10]:=
```
displayWorldSpot[{0, color_}, opts___] := Graphics[
    Hue[0],
    Background → (ColoringFunction /. {opts} /. Options[displayWorldSpot])[
       BackgroundHue /. {opts} /. Options[displayWorldSpot]]
   ];
displayWorldSpot[{1, color_}, opts___] := butterflyGraphics[color, opts];
```

Draw one butterfly.

In[12]:=
```
butterflyGraphics[color_, opts___] := Block[
    {p, q, colFunc},
    colFunc = ColoringFunction /. {opts} /. Options[butterflyGraphics];
    Graphics[
     {
      colFunc[color],
      Polygon[p = {{0, 0}, {1, 3}, {5, 0}, {5, 12}, {1, 8}, {0, 9}}],
      Polygon[q = ({-1, 1} #1 &) /@ p],
      GrayLevel[0.0],
      Line[p],
      Line[q],
      Line[{{0, 9}, {-2, 12}}],
      Line[{{0, 9}, {2, 12}}]
     },
     Background →
      colFunc[BackgroundHue /. {opts} /. Options[butterflyGraphics]],
     PlotRange → {{-6, 6}, {-1, 13}}
    ]
   ];
```

Generate all butterflies

In[13]:=
```
displayButterflyWorld[buts_List, opts___] := Block[
    {dispMatrixWidth},
    dispMatrixWidth =
     DisplayMatrixWidth /. {opts} /. Options[displayButterflyWorld];
    If[dispMatrixWidth === 0, dispMatrixWidth = Length[buts]];
    Show[
     GraphicsGrid[
      Partition[(displayWorldSpot[#1, opts] &) /@ buts, dispMatrixWidth]
     ]
    ]
   ];
```
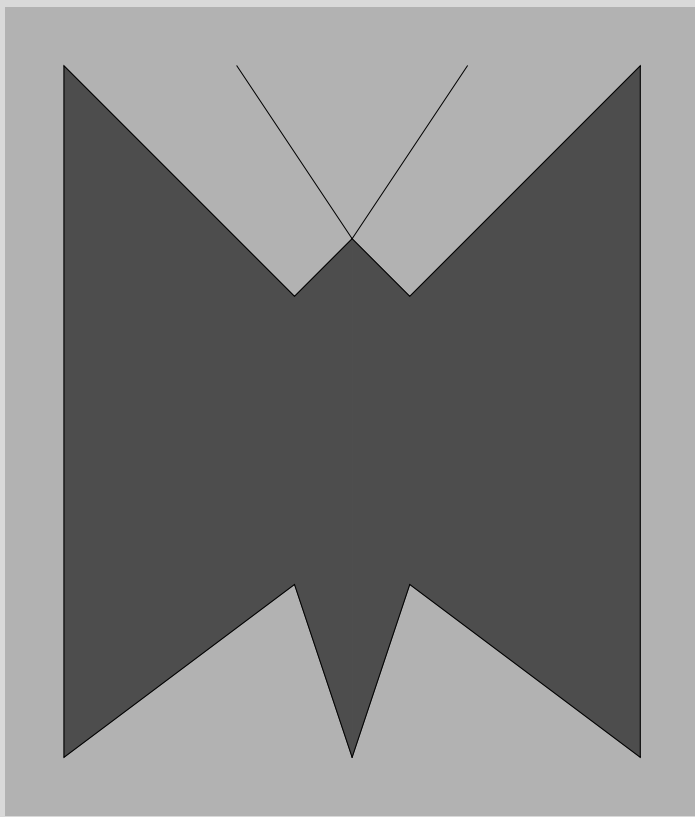
Test

Test displayWorldSpot function.

In[14]:=
```
displayWorldSpot[{0, .8}, BackgroundHue → 0.4]
```

Out[14]=

In[15]:=
```
displayWorldSpot[{1, 0.3}, BackgroundHue → 0.7]
```
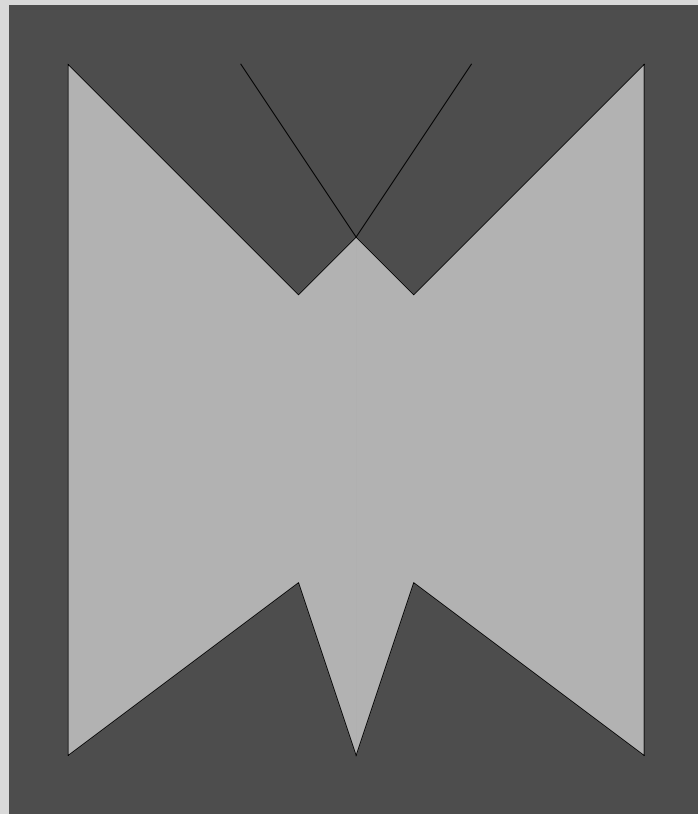
Out[15]=

Test butterflyGraphics function.

In[16]:= **butterflyGraphics[0.7, BackgroundHue → 0.3]**
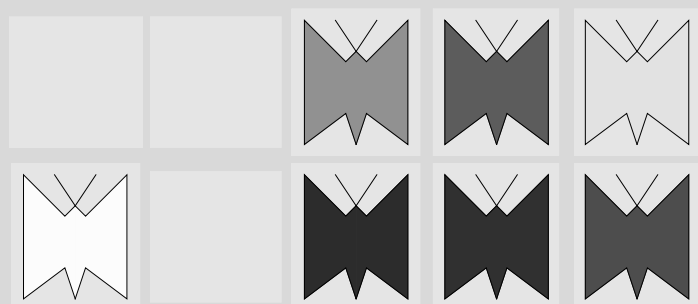
Out[16]=

Test displayButterflyWorld function.

In[17]:= **displayButterflyWorld[butterflies]**

Out[17]=

In[18]:= **displayButterflyWorld[butterflies, DisplayMatrixWidth → 5]**

Out[18]=

Fitness and Misfitness

The fitness of a butterfly is calculated as:
$$\text{fitness} = \frac{1}{butterfly\_color - background\_color}$$
and the misfitness is:

misfitness = butterfly_color - background_color.
The larger the misfitness, the higher probability the butterfly will be removed from the tree trunk (ie. eaten by a bird).

In[19]:=
```
butterflyMisfitness[color_, bgColor_] := Abs[color - bgColor]
butterflyMisfitness[BGColor, _] := 0
```

Test

Test butterflyMisfitness function.

In[21]:=
```
misfits = butterflyMisfitness[Last[#], 0.9] & /@ Cases[butterflies, {1, _}]
```

Out[21]=
```
{0.329755, 0.539946, 0.0154116, 0.0876479, 0.728949, 0.710559, 0.598305}
```

In[22]:=
$$\text{fits} = \left(\frac{\#}{\text{Plus @@ misfits}} \& \right) \text{/@ misfits}$$

Out[22]=
```
{0.109532, 0.17935, 0.00511915, 0.0291134, 0.24213, 0.236021, 0.198735}
```

In[23]:=
```
Total[fits]
```

Out[23]=
```
1.
```

Mutation

We mutate the butterfly's color by adding a random real Δcolor

In[24]:=
```
mutation[{1, c_}, opts___] := Block[
    {mutRange},

    mutRange = MutationRange /. {opts} /. Options[mutation];
    If[RandomReal[] < (MutationProbability /. {opts} /. Options[mutation]),
      {1, Mod[c + RandomReal[{-mutRange, mutRange}], 1.0]},
      {1, c}
      ]
    ];
mutation[{0, c_}, ___] := {0, c};
```

Test

Test mutation function.

In[26]:=
```
mutatedButterflies =
  mutation[#, MutationProbability → 0.4, MutationRange → 0.2] & /@ butterflies
```

Out[26]=
```
{{0, BGColor}, {0, BGColor}, {1, 0.378795}, {1, 0.360054}, {1, 0.0221878},
  {1, 0.998593}, {0, BGColor}, {1, 0.236528}, {1, 0.189441}, {1, 0.365338}}
```

In[27]:= **mutatedButterflies // TableForm**

Out[27]//TableForm=

```
0    BGColor
0    BGColor
1    0.378795
1    0.360054
1    0.0221878
1    0.998593
0    BGColor
1    0.236528
1    0.189441
1    0.365338
```

Now, you can compare it with butterflies.

In[28]:= **butterflies // TableForm**

Out[28]//TableForm=

```
0    BGColor
0    BGColor
1    0.570245
1    0.360054
1    0.884588
1    0.987648
0    BGColor
1    0.171051
1    0.189441
1    0.301695
```

Selection and Evolution

Reduction

Reduction function is used to remove a butterfly from tree trunk. Firstly, we calculate the misfitnesses of all butterflies. Then, randomly select a butterfly. Keep in mind that the higher the misfitness is, the higher probability that the butterfly will be selected and removed is.

In[29]:=
```
reduction[pop_List, opts___] := Block[
    {r, fitnesses},

    bgHue = BackgroundHue /. {opts} /. Options[reduction];
    fitnesses = (butterflyMisfitness[#1, bgHue] &) /@ ((#1[[2]] &) /@ pop);
    ReplacePart[pop, fitnessProportionalSelection[fitnesses] → {0, BGColor}]
    ];
```

Create new butterfly

Once a butterfly is removed, a new butterfly have to be added to the tree trunk. Here, we just randomly select one from the survived butterflies, mutate it and put it on a random place of the tree trunk.

In[30]:=
```
creation[pop_List, opts___] := Block[
    {butPositions, selectedButIndex, nonButPositions, selectedNonButIndex},

    butPositions = Flatten[Position[pop, {1, _}]];
    selectedButIndex = randomSelection[butPositions];

    nonButPositions = Flatten[Position[pop, {0, _}]];
    selectedNonButIndex = randomSelection[nonButPositions];

    ReplacePart[pop,
      selectedNonButIndex → mutation[pop[[selectedButIndex]], opts]]
  ];
```

Fitness Proportional Selection

Select a butterfly according to the misfitness.

In[31]:=
```
fitnessProportionalSelection[fits_List] := Block[
    {randpos, members = Length[fits],
     totalsum = Total[fits], partsum = First[fits], index = 1},

    randpos = RandomReal[{0, N[totalsum]}];
    (*NestWhile[#+=fits[[index]]&,partsum,partsum<randpos&&index++];
    Return[index]*)
    While[partsum < randpos && index < members, index = index + 1;
     partsum = partsum + fits[[index]];
    ];
    index
  ];
```

Auxiliary Functions

In[32]:=
```
randomSelection[elements_List] :=
  elements[[RandomInteger[{1, Length[elements]}]]]
```

In[33]:=
```
creationIterated[pop_List, iter_Integer: 1, opts___] :=
  Nest[creation[#1, opts] &, pop, iter];
```

In[34]:=
```
reductionIterated[pop_List, iter_Integer: 1, opts___] :=
  Nest[reduction[#1, opts] &, pop, iter];
```

In[35]:=
```
reductionAndCreation[pop_List, indivs_Integer: 1, opts___] :=
  creationIterated[reductionIterated[pop, indivs, opts], indivs, opts];
```

Test

Test fitnessProportionalSelection function.

In[36]:=
```
fitnessProportionalSelection[misfits]
```

Out[36]=
```
2
```

Test reduction function.

In[37]:=
```
reductionResults = reduction[butterflies]
```

Out[37]=
```
{{0, BGColor}, {0, BGColor}, {1, 0.570245}, {1, 0.360054}, {1, 0.884588},
  {1, 0.987648}, {0, BGColor}, {1, 0.171051}, {0, BGColor}, {1, 0.301695}}
```

Test creation function.

In[38]:=
```
creationResults = creation[reductionResults]
```

Out[38]=
```
{{0, BGColor}, {0, BGColor}, {1, 0.570245}, {1, 0.360054}, {1, 0.884588},
  {1, 0.987648}, {0, BGColor}, {1, 0.171051}, {1, 0.171051}, {1, 0.301695}}
```

Now, you can compare butterflies, reductionResults and creationResults.

In[39]:=
```
butterflies // TableForm
```

Out[39]//TableForm=
```
0    BGColor
0    BGColor
1    0.570245
1    0.360054
1    0.884588
1    0.987648
0    BGColor
1    0.171051
1    0.189441
1    0.301695
```

In[40]:=
```
reductionResults // TableForm
```

Out[40]//TableForm=
```
0    BGColor
0    BGColor
1    0.570245
1    0.360054
1    0.884588
1    0.987648
0    BGColor
1    0.171051
0    BGColor
1    0.301695
```

In[41]:= **creationResults // TableForm**

Out[41]//TableForm=

```
0    BGColor
0    BGColor
1    0.570245
1    0.360054
1    0.884588
1    0.987648
0    BGColor
1    0.171051
1    0.171051
1    0.301695
```

In[42]:= **displayButterflyWorld[butterflies]**

Out[42]=

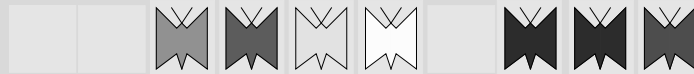In[43]:= **displayButterflyWorld[reductionResults]**

Out[43]=

In[44]:= **displayButterflyWorld[creationResults]**

Out[44]=

Main Function ----- Evolution of Butterflies

In[45]:=
```
evolveButterflies[opts___] := Block[
    {initPop, reproIndivs, indivs, butterflyGenerations},

    initPop = InitialPopulation /. {opts} /. Options[evolveButterflies];
    If[initPop == {},
     initPop = createWorld[
       WorldWidth /. {opts} /. Options[evolveButterflies],
       ButterflyProbability /. {opts} /. Options[evolveButterflies],
       opts
      ]
    ];

    indivs = Count[initPop, {1, _}];

    reproIndivs = Round[
      Length[initPop] ReductionCreationPortion /. {opts} /.
       Options[evolveButterflies]
     ];

    butterflyGenerations = NestList[
      reductionAndCreation[#1, Min[reproIndivs, indivs], opts] &,
      initPop,
      Generations /. {opts} /. Options[evolveButterflies]
     ];

    (*(displayButterflyWorld[#1,opts]&)/@butterflyGenerations*)

    Return[butterflyGenerations]
   ];
```

Test

Test evolveButterflies function.

In[46]:=
```
butterflies2D = evolveButterflies[
  InitialPopulation → {},
  WorldWidth → 20,
  ButterflyProbability → .8,
  ReductionCreationPortion → 0.5,
  Generations → 10,
  MutationProbability → .2,
  MutationRange → 0.5,
  BackgroundHue → 1.,
  ColoringFunction → Hue,
  DisplayMatrixWidth → 5
]
```

Out[46]=

```
{{{1, 0.51611}, {1, 0.889086}, {1, 0.905196}, {0, BGColor}, {0, BGColor},
   {1, 0.68658}, {1, 0.803583}, {1, 0.986423}, {1, 0.605705}, {1, 0.276419},
   {1, 0.346702}, {1, 0.990982}, {1, 0.641313}, {0, BGColor}, {1, 0.902987},
   {0, BGColor}, {1, 0.21259}, {1, 0.038618}, {1, 0.498931}, {1, 0.228424}},
  {{1, 0.641313}, {1, 0.123709}, {1, 0.905196}, {1, 0.641313}, {1, 0.605705},
   {0, BGColor}, {1, 0.605705}, {1, 0.986423}, {1, 0.605705}, {1, 0.276419},
   {0, BGColor}, {1, 0.990982}, {1, 0.641313}, {1, 0.641313}, {1, 0.641313},
   {1, 0.641313}, {0, BGColor}, {0, BGColor}, {1, 0.605705}, {1, 0.82404}},
  {{1, 0.990982}, {1, 0.376373}, {1, 0.990982}, {0, BGColor}, {1, 0.605705},
   {1, 0.990982}, {1, 0.605705}, {1, 0.986423}, {1, 0.605705}, {0, BGColor},
   {1, 0.990982}, {1, 0.990982}, {1, 0.605705}, {1, 0.641313}, {0, BGColor},
   {1, 0.986423}, {0, BGColor}, {1, 0.641313}, {1, 0.990982}, {1, 0.82404}},
  {{1, 0.986423}, {1, 0.986423}, {1, 0.990982}, {0, BGColor}, {1, 0.986423},
   {1, 0.990982}, {1, 0.975004}, {1, 0.986423}, {0, BGColor}, {1, 0.358901},
   {0, BGColor}, {1, 0.990982}, {1, 0.986423}, {1, 0.986423}, {1, 0.986423},
   {1, 0.986423}, {1, 0.986423}, {0, BGColor}, {1, 0.990982}, {1, 0.47623}},
  {{1, 0.881057}, {1, 0.986423}, {1, 0.990982}, {1, 0.211739}, {1, 0.986423},
   {1, 0.990982}, {0, BGColor}, {1, 0.211739}, {0, BGColor}, {1, 0.881057},
   {1, 0.986423}, {1, 0.211739}, {1, 0.986423}, {1, 0.211739}, {0, BGColor},
   {1, 0.986423}, {1, 0.986423}, {1, 0.990982}, {1, 0.990982}, {0, BGColor}},
  {{1, 0.986423}, {1, 0.986423}, {1, 0.990982}, {1, 0.664164}, {1, 0.986423},
   {0, BGColor}, {1, 0.388904}, {1, 0.986423}, {1, 0.986423}, {1, 0.986423},
   {1, 0.986423}, {1, 0.986423}, {1, 0.986423}, {1, 0.986423}, {0, BGColor},
   {1, 0.986423}, {0, BGColor}, {1, 0.5463}, {1, 0.990982}, {0, BGColor}},
  {{1, 0.986423}, {1, 0.670326}, {1, 0.990982}, {0, BGColor}, {1, 0.708436},
   {1, 0.986423}, {1, 0.990982}, {0, BGColor}, {1, 0.986423}, {0, BGColor},
   {1, 0.986423}, {0, BGColor}, {1, 0.986423}, {1, 0.986423}, {1, 0.0785392},
   {1, 0.171483}, {1, 0.986423}, {1, 0.0785392}, {1, 0.0214506}, {1, 0.986423}},
  {{1, 0.986423}, {0, BGColor}, {0, BGColor}, {0, BGColor}, {0, BGColor},
   {1, 0.986423}, {1, 0.990982}, {1, 0.986423}, {1, 0.887176}, {1, 0.986423},
   {1, 0.141714}, {1, 0.986423}, {1, 0.986423}, {1, 0.196427}, {1, 0.001773},
   {1, 0.986423}, {1, 0.986423}, {1, 0.986423}, {1, 0.196427}, {1, 0.986423}},
  {{1, 0.500445}, {0, BGColor}, {1, 0.500445}, {0, BGColor}, {1, 0.986423},
   {1, 0.986423}, {1, 0.990982}, {1, 0.986423}, {1, 0.990982}, {0, BGColor},
   {1, 0.901194}, {1, 0.986423}, {1, 0.901194}, {1, 0.986423}, {0, BGColor},
   {1, 0.986423}, {1, 0.990982}, {1, 0.986423}, {1, 0.679511}, {1, 0.986423}},
  {{1, 0.990982}, {1, 0.990982}, {1, 0.990982}, {1, 0.990982}, {1, 0.990982},
   {1, 0.986423}, {1, 0.990982}, {1, 0.986423}, {1, 0.990982}, {0, BGColor},
   {1, 0.986423}, {1, 0.986423}, {0, BGColor}, {1, 0.544177}, {0, BGColor},
   {1, 0.986423}, {1, 0.990982}, {0, BGColor}, {1, 0.990982}, {1, 0.990982}},
  {{1, 0.990982}, {1, 0.254934}, {1, 0.990982}, {0, BGColor}, {1, 0.990982},
   {1, 0.633884}, {1, 0.990982}, {1, 0.681481}, {1, 0.633884}, {1, 0.633884},
   {1, 0.169822}, {1, 0.990982}, {0, BGColor}, {1, 0.633884}, {1, 0.731436},
   {0, BGColor}, {1, 0.990982}, {0, BGColor}, {1, 0.990982}, {1, 0.731436}}}
```
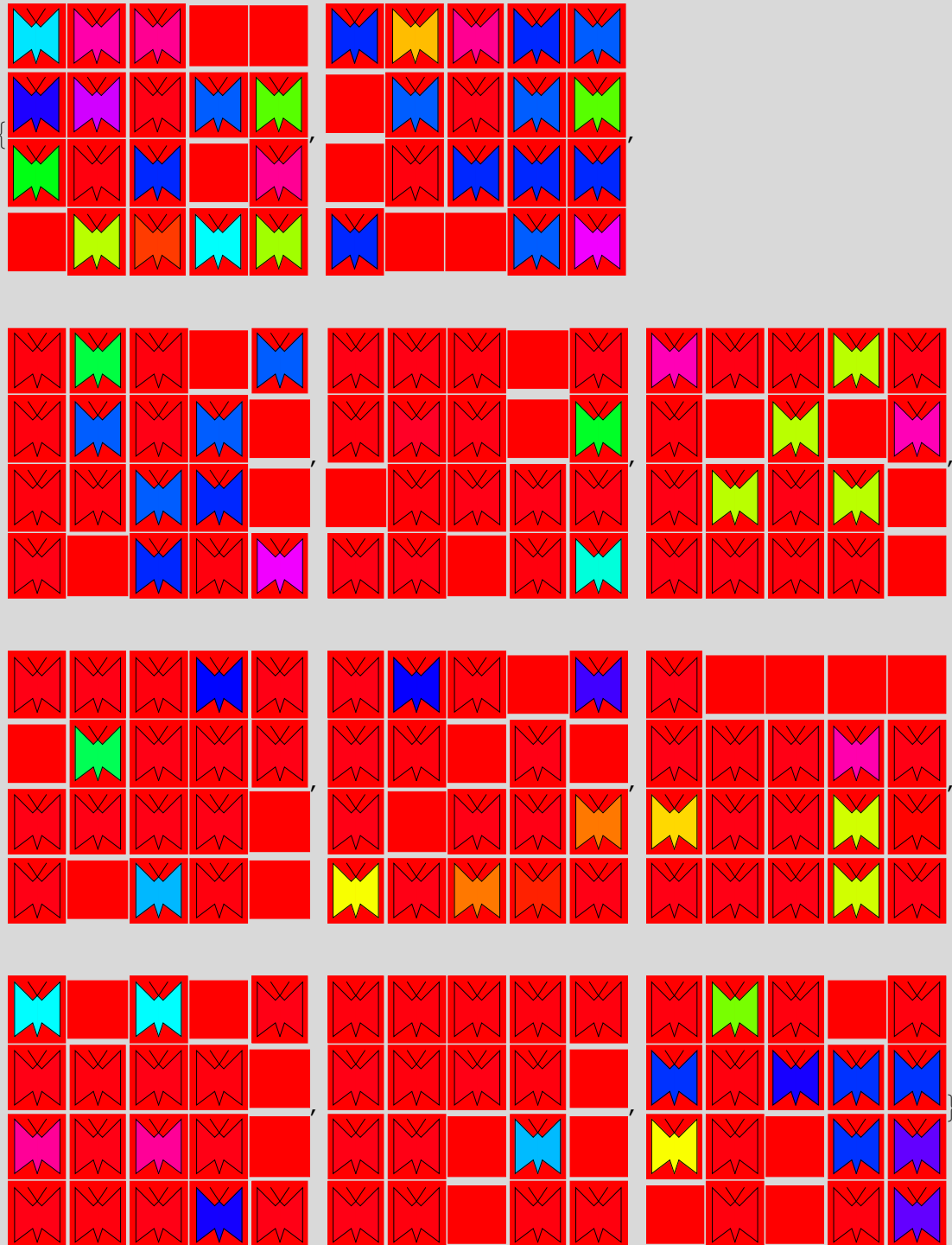
Now, we display each generation in graphics.

In[47]:=
```
displayButterflyWorld[#, BackgroundHue → 1.0,
    ColoringFunction → Hue, DisplayMatrixWidth → 5] & /@ butterflies2D
```

Out[47]=

Experimentation

Butterflies on red tree trunk

In[48]:=
```
butterfliesOnRedTree = evolveButterflies[
    InitialPopulation → {},
    WorldWidth → 50,
    ButterflyProbability → .8,
    ReductionCreationPortion → 0.5,
    Generations → 15,
    MutationProbability → .2, MutationRange → 0.5,
    BackgroundHue → 1.,
    ColoringFunction → Hue,
    DisplayMatrixWidth → 10
    ];
```
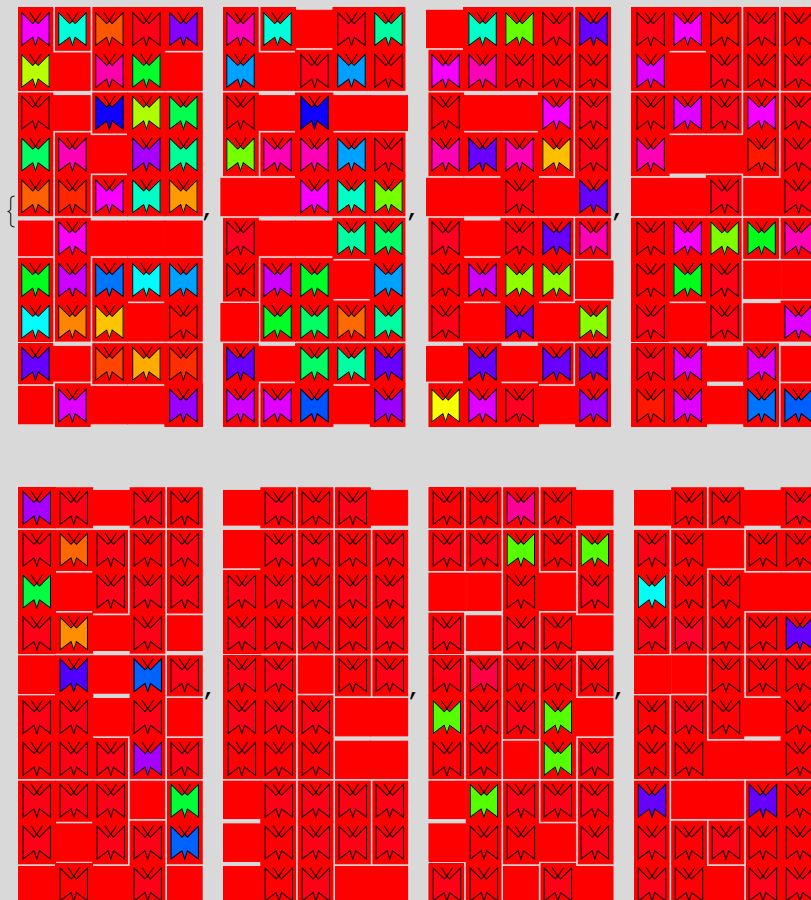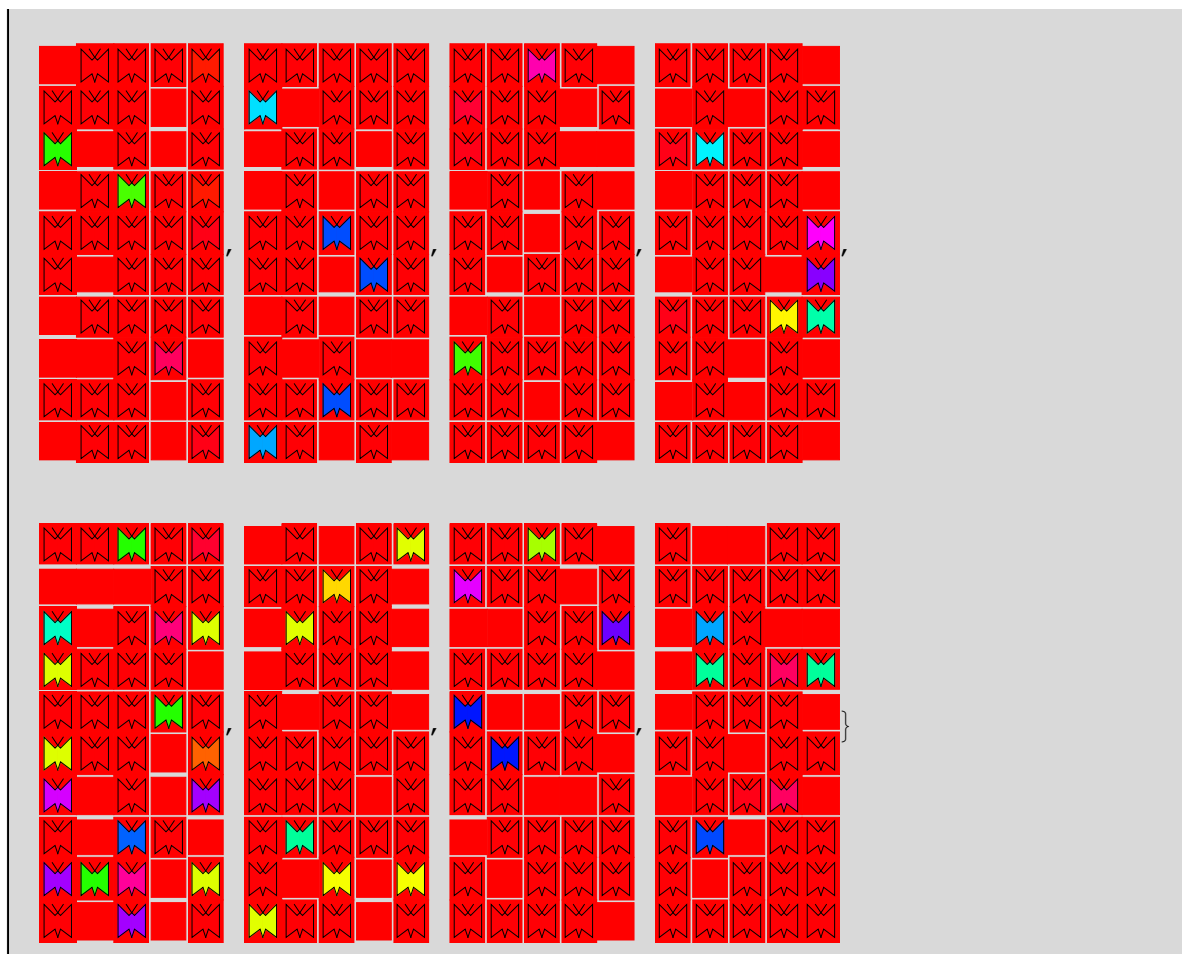
In[49]:=
```
displayButterflyWorld[#, BackgroundHue → 1.0,
    ColoringFunction → Hue, DisplayMatrixWidth → 5] & /@ butterfliesOnRedTree
```

Out[49]=

Butterflies on cyan tree trunk

```
In[50]:=  butterfliesOnCyanTree = evolveButterflies[
            InitialPopulation → Last[butterfliesOnRedTree],
            WorldWidth → 50,
            ButterflyProbability → .8,
            ReductionCreationPortion → 0.5,
            Generations → 15,
            MutationProbability → .2, MutationRange → 0.5,
            BackgroundHue → .5,
            ColoringFunction → Hue,
            DisplayMatrixWidth → 10
          ];
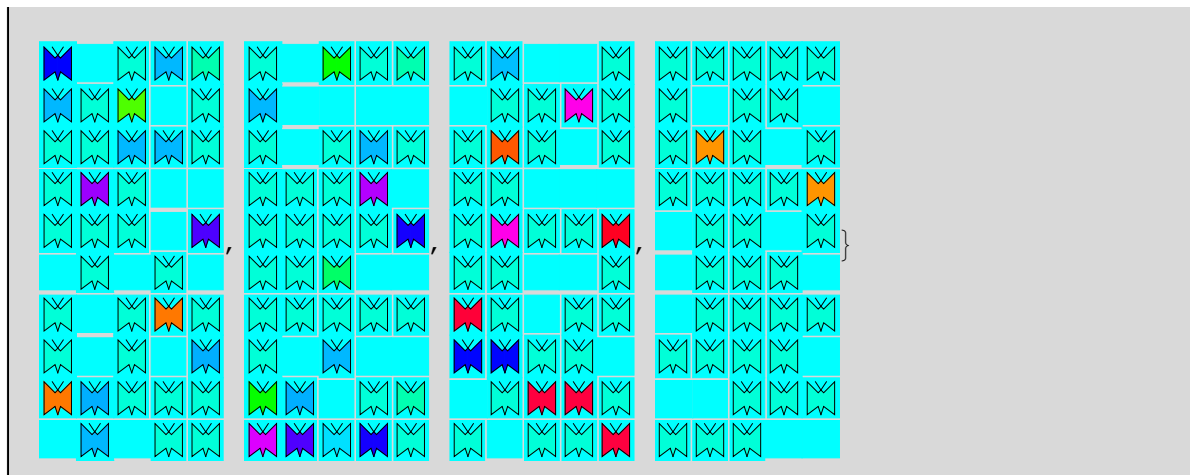```

```
In[51]:=  displayButterflyWorld[#, BackgroundHue → 0.5, ColoringFunction → Hue,
            DisplayMatrixWidth → 5] & /@ butterfliesOnCyanTree
```

Out[51]=

Butterflies on light green background

```
In[52]:=  butterfliesOnGreenTree = evolveButterflies[
            InitialPopulation → Last[butterfliesOnCyanTree],
            WorldWidth → 50,
            ButterflyProbability → .8,
            ReductionCreationPortion → 0.5,
            Generations → 15,
            MutationProbability → .2, MutationRange → 0.5,
            BackgroundHue → .2,
            ColoringFunction → Hue,
            DisplayMatrixWidth → 10
          ];
```
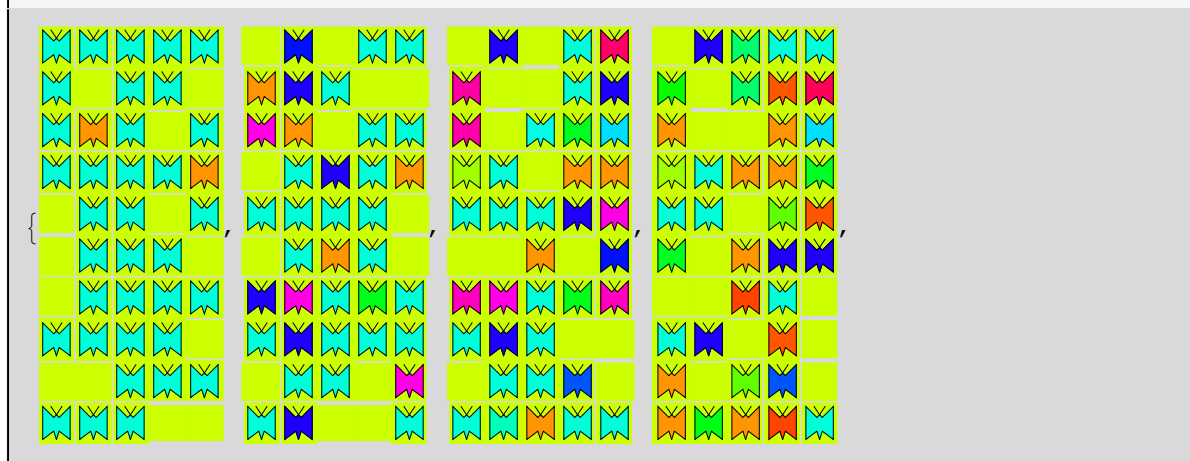
```
In[53]:=  displayButterflyWorld[#, BackgroundHue → 0.2, ColoringFunction → Hue,
            DisplayMatrixWidth → 5] & /@ butterfliesOnGreenTree
```

Out[53]=

Save Data

In[54]:=
```
path = "C:\\Users\\zbche_000\\OneDrive\\Documents\\Technical
    Documents\\Learning Mathematica\\Coder\\Illustrating
    Evolutionary Computation with Mathematica by Christian Jacob\\";
SetDirectory[path <> "Chapter 01\\Butterflies\\"];
butterflies2D >> "Butterflies2D"
butterfliesOnRedTree >> "ButterfliesOnRedTree"
butterfliesOnCyanTree >> "ButterfliesOnCyanTree"
butterfliesOnGreenTree >> "ButterfliesOnGreenTree"
```