All cells are dead in two successive patterns. It is then assumed that the survival rules will not generate new live cells, and one of the options below come into effect.

The original pattern is loaded and the survival rules are reset to their initial values. The patterns that follow will thus duplicate those that came before.

The original pattern is loaded, and the survival rules are mutated.

The survival rules are mutated. With this option selected, it might take some time before a mutation arises that will generate live cells in the absence of other live ones.

Nothing is changed, and ***Life*** continues to generate the same extinct pattern until it is interrupted. This is obviously not recommended.

All cells are alive in two successive patterns. It is then assumed that the survival rules will not kill any cells, and one of the options below come in effect.

The survival rules are mutated. With logical survival rules in effect, it might take some time before a mutation arises that will kill some cells in the current pattern. If count-based survival rules are used, any mutation will immediately cause some cells to die.

Nothing is changed, and ***Life*** continues to generate the same over populated pattern until it is interrupted. This is obviously not recommended.

Mutations are random changes to the survival rules, be they count-based or logical.

With this option selected, a mutation will be induced at the interval specified. This is recommended.

The number of generations between mutations.

A screen saver is normally interrupted by mouse activity, such as movement or clicking. This will be the case if you select quitting here. If you select any of the other options, clicking the mouse will cause that event to happen, and the screen saver will not be interrupted. Mouse movements will then be ignored.

The screen saver behaves normally. Moving or clicking the mouse interrupts.

The survival rules are mutated when the mouse is clicked.

An epidemic of the selected type is caused by clicking the mouse.

A previous pattern interferes with the current one when the mouse is clicked.

During epidemics certain portions of the current pattern die. You can select both what the shape of the affected portion must be, and what fraction of the total pattern it must cover.

With this option selected, an epidemic of the selected shape and coverage will occur at the interval specified. This is recommended.

The number of generations between epidemics.

The selected shape of the epidemic will be sized such that it covers this percentage of the total pattern.

The shape of the epidemic. This shape will be sized in accordance with the percentage of coverage.

Life will not validate a password (if one is specified) for the number of seconds specified here.

The number of milliseconds between new patterns. If you have specified a path that covers only a small portion of the screen, or comprises only a few patterns, a delay might be needed in order to see the patterns long enough.

With this option selected, ***Life*** will beep whenever a mutation, epidemic or atavism is in effect. This option should generally be off, as extinction often causes long sequences of mutations, which will lead to irritating levels of sound.

Save the configuration, and exit.

General help

Exit without saving.

Count-based survival rules

Survival of each cell in the next pattern is based on the number of live neighbours of that cell in the previous pattern.

Logical survival rules
Survival of each cell in the next pattern is based on the distribution of live cells among the neighbours of that cell in the previous pattern.

If the cell has this many live neighbours in the previous pattern, it will be alive in the next if this option is selected, and dead if it is not.

When neighbours are counted, only these neighbouring cells will be considered. There must be at least 3 neighbouring cells selected here.

A logical distribution of live neighbours is marked here. Any cell surrounded by live neighbours in this shape in the previous pattern will be alive in the next. By exclusion it means that if this distribution is not specified as a logical survival rule, and a cell is surrounded in this manner, that it will die.

Clears the markers of neighbouring cells for easier selection.

Remove the selected logical survival rule from the list.

Add the logical survival rule currently specified to those already listed.

Removes all logical survival rules.

Loads the logical survival rules last saved to the registry.

The logical survival rules specified so far. The rules are represented here with X's and O's, the X's denoting a cell marked as alive, the O's a cell marked as dead, and the string corresponding to the actual structure from top left to bottom right.

The colour a cell will have when it lives due to this survival rule. This colour is based on the number of cells alive in the structure, as specified under colours.

A cell that is alive is drawn with a colour related to the number of live neighbours it had in the previous pattern. These specifications of colour are used when both count-based and logical survival rules are in effect.

A cell that lives in the next pattern due to a survival rule that involves the presence of N live neighbours in the previous pattern, will be drawn with this colour. Click the box to change the color.

Select this option if you want ***Life*** to generate a random survival rule of the type you have selected.

Select this option if the pattern must be deemed to fit on a torus, or doughnut. This is the same as saying that the left edge is adjacent to the right edge, and the top is adjacent to the bottom. Cells in the top row, for example, will then have neighbours above them that are actually in the bottom row.

Survival will be determined by the number of live neighbours of each cell when this option is selected.

Survival will be determined by the structure, or placement of live neighbouring cells when this option is selected.

Over and above the living and dying of cells in terms of the survival rules, you may specify that cells which have been alive for a certain number of generations, must die irrespective of the number or structure of their live neighbours.

Mortality is applied, and the ages of cells are tracked and checked. When a cell has been alive for the specified number of generations, without dying, it dies.

The age of a cell when it dies due to mortality.

The recurrence of certain traits a number of generations later. You have the option of letting the original pattern, or the pattern N generations ago, blend with the current pattern in a number of ways. When atavism is in effect, the cells of the old pattern will be compared with those of the new pattern, one for one, using logical comparison, and the resultant pattern will then represent both the current one as well as the influence of the old.

Do not apply atavism

Cells are compared using AND : If both the old and the current cell are alive, the resultant cell is alive, otherwise it is dead.

Cells are compared using OR : If either the old or the current cell is alive, the resultant cell is alive.

Cells are compared using exclusive OR : If one of the old or current cells is alive, but not both, the resultant cell is alive.

Cells are compared using equivalence : If both the old and the current cell are alive, or both are dead, the resultant cell is alive.

The interval at which atavism is applied.

The cells in the resultant pattern, after atavism has been applied, are all inverted. Those that are dead live, and those that are alive die.

Atavism is applied using the result of the last atavistic interference as the old pattern. If this option is not selected, the original pattern is used throughout.

The screen is represented here, each square the placeholder of a pattern in the sequence of generations that ***Life*** will generate. Draw the path that ***Life*** must use to traverse the screen when the screen saver is active. Use the left button to draw the path, and the right button to remove a square from the path.
The square marked in green is the starting point of the path.

A few predefined paths. Selecting a path here will clear whatever path is currently specified.

The path is traversed starting at its end point and ending at the start.

Select this option if you want ***Life*** to traverse the path both ways, from start to finish and back again.

Selecting this option will clear the screen once a path has been fully traversed. If back-tracking is selected, the path is traversed both ways before the screen is cleared.

Select the animation speed used when viewing the path here. This has no effect on the screen saver itself.

The magnification of the pattern when drawn on-screen.

The width, in pixels, of the space left open between adjacent patterns on the screen.

Selecting this option will ensure that a path is centered on the screen when possible. I you change the Scale, Size or Padding settings, which all affect the number of patterns that can fit the screen, but do not specify a new path, the path last saved is mismatched in terms of the screen when ***Life*** runs. If it is possible this path will then be centered on the screen.

The actual pattern. This pattern is symmetrical around its diagonal running from the top left corner to the bottom right. It represents the upper left quarter of the full pattern displayed on screen when ***Life*** runs, the other three quarters being symmetrical reflections of the actual pattern.
Use the left mouse-button to draw live (black) cells.
Use the right mouse-button to draw dead (white) cells.

The full pattern, comprised of the actual pattern reflected vertically and horizontally.
Use the left mouse-button to draw live (black) cells.
Use the right mouse-button to draw dead (white) cells.

The actual pattern is overlaid with a grid when this option is selected. Turn this option off for faster drawing.

The width of the pen while drawing. If a width greater than 1 is used, you will need to synchronize the actual and full patterns from time to time.

The size of the actual pattern. Changing the size clears the currently selected pattern.

Defaults for all settings are loaded and those last saved to the registry are deleted. The path last specified is not affected by this.

Overview of *Life*

Life is a screen saver that produces patterns in an evolving sequence, using rules you specify and a starting pattern, or seed.
The idea is partly based on what is known as Conway's game of Life, a simulation of cellular automata.

Each pattern is comprised of an array of NxN cells. Each cell has two states: either it is alive or it is dead. Generally, excluding borders, a cell in such a grid, like on a chess board, has eight neighbours. Using a rule that governs survival, and looking only at the neighbours of a given cell in the current pattern, the state of that cell in the new pattern is determined. This is done for every cell in the pattern, and the pattern is then drawn on the screen. The new pattern becomes the current pattern, and the process is repeated.

To make things more interesting the colour with which live cells are drawn can be controlled, and is defined in terms of the number of live neighbours it had in the previous pattern. All dead cells are drawn in white.

The rules of survival can mutate, involving random changes. This is often necessary as some rules, when applied to a pattern, can yield extinction, in which case all cells die and stay dead, or overpopulation, in which case all cells are alive and refuse to die. Mutation is a way out of these situations, which, in terms of a screen saver, are not good news.

Epidemics may be specified to occur at regular intervals. During an epidemic a certain sector of the pattern dies. You may specify the shape of this sector, as well as the portion of the pattern it covers.

Atavism, the recurrence of "genetic" traits of past generations, may also be allowed. If this is the case, a pattern generated a certain number of generations ago, or the original pattern, will cross-breed with the current pattern, and thus alter the shape of those to come. The way in which the two patterns are merged can be controlled.

Mortality may additionally be applied to cells, keeping track of their ages, and killing them if they grow too old.

You have the option of disabling the mouse in the sense of its ability to interrupt the screen saver. If this is done, clicking the mouse can cause mutations, epidemics or atavism to occur while you watch, and when you want.

Lastly, the way in which ***Life*** displays successive patterns on the screen can be controlled through the creation of Paths. The scale with which the pattern is drawn, and the width of open padding between adjacent patterns can be controlled. In terms of these settings the screen is comprised of a rectangular virtual mesh of square spaces, wherein the patterns may be drawn. The dimensions of this mesh are further influenced by the resolution of your screen.
The order in which patterns are drawn in these virtual place-holders is called a path. In that way you can control whether ***Life*** displays successive patterns across and then down, for example, as opposed to a spiral hop-scotching path that excludes certain areas of the screen.

Patterns and Neighbours

The pattern is the square array of cells that represents the world in which the evolution of ***Life*** happens. Since rotational symmetry is enforced, only a quarter, the upper left quarter, of the whole is stored. This quarter is in itself symmetric around its diagonal.
The pattern has dimensions ranging from 20x20 (yielding a full pattern of 40x40) to 64x64. To start the evolution off, you may draw an initial pattern. In so doing you create cells that are alive, and leave others dead. When ***Life*** runs, your initial pattern is used to generate the second pattern, using the specified survival rules, the second is used to generate the third, and so on.

You may specify the pattern to be toroidal, in which case the left edge is deemed to be joined to the right, and the top edge to the bottom. In that way a cell in the top row will have neighbours "above" it that are actually in the bottom row, and so on.

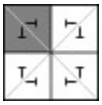


Figure 1

In *Figure 1* above the whole is shown as the reflections, horizontally to the right, vertically down and horizontally and then vertically to the bottom right corner, of the actual pattern, which is shaded. The actual pattern is also symmetrical along its diagonal as shown.

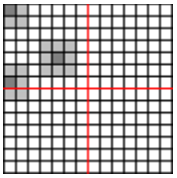


Figure 2

In *Figure 2* three cells are shown, and their neighbours have been lightly shaded. In this case the pattern is not toroidal, and the cell in the upper left corner thus has only the neighbours shown. It is also important to note that the cell on the horizontal middle line has neighbours *outside* the pattern, and that this is the case even in the absence of toroidal behaviour, because those cells in the top row of the bottom quarter are reflections of the cells in the bottom row of the top quarter.

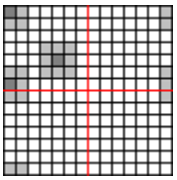


Figure 3

In *Figure 3* the same pattern is shown as in Figure 3, but with toroidal behaviour.

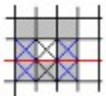


Figure 4

In *Figure 4* above is an illustration of neighbourhood behaviour on the internal horizontal edge, between a

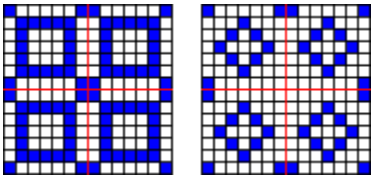
top quarter and its bottom reflection. The white cell has the neighbours indicated by shading, but one of its neighbours is in fact its own reflection, the cell directly beneath it. It will thus always be true that a cell on such a border, or the vertical equivalent, has at least 1 live neighbour if it is alive itself, namely its own reflection.

Survival Rules

Count-based Survival Rules

When count-based rules apply, the following process is undertaken for each cell in the pattern.

- Its live neighbours are counted, taking toroidal behaviour into account if its applies.
- If you have specified that a cell with that number of live neighbours (a number between 0 and 8) must live, the cell in question is alive in the next pattern. If you did not, it is dead.
- If the cell is alive in the next pattern, it is drawn in the colour specified for that number of live neighbours. This colour may be any colour that the system can display, except for white, which is used by ***Life*** for dead cells.



Survival rules 1

In *Survival rules 1* above, a fictitious 7x7 pattern with initial cells are moved one generation forward using the count-based survival rule (1,2,5). In other words, all cells with 1, 2 or 5 live neighbours lived, the others died. Because there are 9 possible counts that you may toggle, 0 through 8, there is a total of 512 different count-based survival rules.

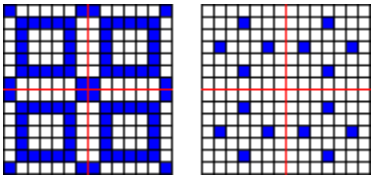
Note
You may specify that only certain of the neighbours of a cell be considered when counting its live neighbours. This does not affect the nature of the process, only the possible outcomes.

Logical Survival Rules

Logical survival rules are based on the specification of structures by which the cell must be surrounded if it is to live. An example would be to say that cells having live neighbours above, below, to the left and to the right, but in no other position, must live. To this can then be added the specification that a cell with one live neighbour above it and to the left, must live. As there are 8 positions surrounding a cell, there are 256 possible structures surrounding it.

Survival rules 2

In *Survival rules 2* above, 4 logical survival rules are depicted. If you had specified these as your logical survival rules, all cells surrounded like that will live, and all others will die.



Survival rules 3

In *Survival rules 3* above, a fictitious 7x7 pattern with live cells is taken forward one generation with the first two of the logical survival rules depicted in *Survival rules 2*.

Colours

When a live cell is drawn, it is drawn using a colour that relates to its number of live neighbours. This is the case even if logical survival rules apply.

These colours are specified during configuration.
In the case of count-based survival rules, there is a one-to-one correspondence between the colours specified and the live/dead state of a cell with such a number of neighbours.
In the case of logical survival rules, there is no such one-to-one correspondence, as a structure might have the same number of cells in it as another, and if they are both specified, the cells that survive due to them would be drawn in the same colour.

Note
As mutations might occur, rules will be in effect that you did not specify, and it is therefore important that you choose colours for all the numbers of neighbours. As an example, if you used count-based survival rules, and specified that all cells having 1, 2 or 5 live neighbours must live, and you only specified the colours for 1, 2 and 5, a subsequent mutation might cause cells with 6 live neighbours to live, and these will then be drawn in the default colour.

Mutations

A mutation is a random change in the survival rule.

In the case of count-based survival rules, a number between 0 and 8 is generated randomly, and the live/dead state of a cell with that number of live neighbours is toggled, i.e. turned on if it were off, and vice versa.

In the case of logical survival rules, a number between 0 and 255 is generated randomly, and the structure denoted by that number is selected, if it were not, or un-selected if it were.
This is possible because the logical survival rules are enumerated as follows: The neighbouring cells are labelled from top left to bottom right with the numbers 0 to 7. A given structure, which has, say, the corners cells marked, is then denoted by the number 165, it being the sum of 2^0, 2^2, 2^5, and 2^7 (read 2 to the power of 7). No other structure yields this number.

Mutations make the evolution of a pattern more interesting, and, moreover, is required when extinction or overpopulation occurs.
When extinction occurs, it means that all cells have died and remained dead. There will therefore be no more live cells if the current rule is kept the way it is. A mutation is a way out of this predicament.
Similarly, when the pattern is over populated, namely such that all cells are alive and none die, a mutation is again required.

Note
When all cells are extinct, and mutation does not rectify the matter, ***Life*** automatically brings to life the corner cells of the pattern. This is outside of your control, and is such by design. As a screen saver, it will do no good if a lot of white squares fill the screen and remain that way. The same thing happens in reverse if there is overpopulation that mutation cannot rectify.

Mutations may be set to occur at regular intervals during configuration.

Epidemics

An epidemic is an occurrence during which a portion of the pattern is killed. You may choose the shape that this portion must have, and the fraction of the pattern's surface that it must cover. There are essentially five different shapes for an epidemic:

- Outer rim-square rim on the outside of the pattern dies
- Inner rim-square at the center of the pattern dies.
- Straight cross-a cross composed of a vertical and horizontal bar, centered in the pattern, dies.
- Diagonal cross-a cross comprised of the two diagonals of the pattern dies
- Random-random cells totaling the fraction of the surface specified, die.
- Choose-randomly selects one of the above every time that an epidemic occurs.

You may set the type of epidemic during configuration, and the interval at which it occurs.

Atavism

Atavism, in genetics, is the occurrence, generations after the fact, of a trait or peculiarity that is not seen in the intermediate phases of development.

In ***Life***, atavism is the *merging* of an older pattern with the current one.

This *merging* is a cell-for-cell comparison between the older pattern and the current pattern, and may be any of the following:

- AND - Only where both cells are alive does the resultant cell live.
- OR - When either cell is alive, the resultant cell lives.
- XOR - When one of the two cells is alive, but not both, the resultant cell is alive.
- EQV-When both cells are in the same state, alive or dead, the resultant cell is alive.

You may select atavism to invert, which means that the whole pattern, after the merge, is inverted. Live cells become dead cells, and vice versa.

If atavism is progressive, the older pattern is a constant number of generations behind the current pattern, this number being the interval at which atavism occurs. If atavism is not progressive, the older pattern is the original pattern at all times.

You may set the interval at which atavism occurs during configuration.

Mortality

Mortality is a way of killing off cells that have been alive for a number of generations, over and above the rules that govern survival.
With mortality in effect, the ages of all cells are tracked, and those that reach dying age are then killed.
The age of a cell is the number of times it has been alive in successive patterns.

You may set the age at which cells die during configuration.

Stationary patterns and Cycles

Stationary patterns

A stationary pattern is one that does not change when the survival rule is applied to it. Such a pattern will then persist until such time as the survival rule mutates. Special cases of stationary patterns are extinction and overpopulation.

In the case of extinction, all cells are dead, and the rule forces a cell that is fully surrounded by other dead cells, to die. In the next pattern, all cells would again be dead, and the state of extinction would continue. You have the option to guard against this with forced mutation, and other options. These include starting over with the initial pattern.

Overpopulation, on the other hand, is a stationary pattern of cells that are all alive, combined with a rule that keeps a cell alive if it is fully surrounded by live cells. Again, the state would persist, and it can be guarded against with forced mutation and other settings.

Note
If the pattern is not toroidal, the above cases are less likely to occur, as cells on the edges or in the corners of the pattern have a different number of neighbours to those in the interior, and other elements of the survival rule apply to them.

Of greater interest are stationary patterns which do not fall in either of the above classes. These you will come across from time to time, and they are entirely the result of the interaction between the survival rule, and the distribution of live cells within the pattern. If no mutation, epidemic or atavism is allowed to interfere with the pattern, it will persist. For this reason it is of some importance that these effects be allowed in order to have a screen saver.

Cyclic patterns

A cyclic pattern produces itself somewhere in its descendants. In other words, when the survival rule is applied to it, a series of patterns result which, at some point, again produce the starting pattern. The number of generations separating the pattern and its recurrence is called the period of the cyclic pattern. In this way a stationary pattern is also cyclic, with a period of zero.

Extinct and over populated patterns can be cyclic. Imagine, as an example, a count-based survival rule that states, among other things, that fully surrounded cells must die, and that cells with no live neighbours must live. If the pattern reaches extinction the next pattern would be over populated. The pattern after that would be extinct again.

As with stationary patterns, mutations, epidemics and atavism are ways to break the cycle that would otherwise persist.

Want to investigate?

If you want to have a better understanding of stationary or cyclic patterns, do the following. Set ***Life*** to have no mutation, epidemics or atavism. De-activate mortality as well. Specify that mouse-clicks must cause a mutation, for example.
The effect would be that you will be able to watch cycles develop, and repeat. No mutation or other effects will interfere with this behaviour, until such time as you feel like injecting change by clicking the mouse.

Paths

When ***Life*** generates successive patterns, it has to draw these at locations on the screen. A path is a specification of how ***Life*** must do this.

In terms of the scale, pattern size and the padding width, there is a virtual grid of NxM square positions available on the screen. It is virtual because these positions need not all be used. By default, ***Life*** draws patterns from the top left position to the bottom right, in columns, using all the available positions.

When you specify a path, you are telling ***Life*** which position to use first, then the next, and so on. This information is saved to a file life.pth, and read when ***Life*** starts up.

You may also specify that the path is used in reverse, or back-tracked. In the latter situation, ***Life*** will traverse the path from start to finish, and then back again.

Note
If you have specified a path and subsequently changed the settings of scale, padding, or the size of the pattern, the saved path no longer applies to the virtual grid. In this case ***Life*** will use the part of the path that fits the grid, or the default if all of the path falls outside the grid.

Centering the path tells ***Life*** to center the path in the context of the screen, if this is possible.
You may clear the screen between the traversals of a path. ***Life*** will clear the screen after every 50 traversals regardless of this setting.

How to draw a pattern

If you have previously drawn a pattern, and closed the configuration dialog with OK, this pattern is saved to the registry. You may load it at any time, using the load button on the drawing toolbar. The size will be set to the size of the pattern that was saved.

The full pattern is represented at actual size, with a scale of 1, and the top left quarter of the full pattern is represented enlarged. You may draw in either of these views using the drawing tools.

Throughout, black represents live cells, and white dead cells.

Drawing points
Use the left mouse-button to draw in black, and the right button to draw in white.

Drawing lines
Use the left mouse-button for black lines, and the right button for white lines. Position the pointer where you want to start drawing the line, then move it to the end-point while holding the mouse-button down. Releasing the mouse-button draws the line.
If you release the pointer over the starting point while drawing a line, no line will be drawn. You must then click in the picture once to reset line drawing mode.

Filling areas
Use the left mouse-button to fill with black, and the right button to fill with white.
You can only fill with black if the point under the cursor is white, and filling will then extend to the black borders of the region, if there are any. The same holds for filling with white: The point under the cursor must be black.

When the pen width is greater than 1, the full-view and the quarter grow out of sync as drawing progresses. Synchronize the two views using the toolbar button.

Note
If, during configuration, you draw a pattern, or even clear one that you loaded, the new pattern is saved when you exit. If you do not alter the loaded pattern, or do not even load it, no new pattern is saved, and the pattern saved in an earlier session will still be in effect.

How to create a path

Set the scale of drawing, and the padding between adjacent positions. The virtual screen will show the positions available for a path.

Load a previously saved path using the load button on the toolbar. Alternatively, you may specify a new path by clearing the current path.

Note:
Loading a path reads the previously saved path and attempts to fit it into the current virtual screen. Those positions outside the current screen are not used.

Now proceed as follows:

- Use the left mouse-button to click, or drag in the virtual screen. The positions will be selected as you proceed.
- To remove a position from the path, use the right mouse-button.

The starting position of the path is marked with a green dot. You may view the last position specified by pressing the button on the toolbar.
To view the path, use the animate button on the toolbar. This animation is done at a speed set by the animation speed slider.

Further settings:

- If you would like the path you have specified to be used in reverse instead, specify use reversed.
- If you would like the path to be followed from start to finish, and then back again, specify back-track.
- If you would like ***Life*** to clear the screen between traversals, specify clear screen.

When closing the configuration dialog, the specified path, if not empty, is saved.

Speed considerations

The following is a list of aspects, and how they affect the speed at which ***Life*** runs.
Those marked with an * affect the screen saver noticeably.

- Pattern size (bigger=slower) *
- Logical survival rules (=slower) *
- Count-based survival rules (counting only certain neighbours=slower) *
- Mortality (=slower)
- Toroidal pattern (=slower)
- Mutation (smaller interval=slower)
- Epidemics (smaller interval=slower)
- Epidemics (diagonal=slower)
- Atavism (smaller interval=slower)
- Atavism (EQV=slower)
- Atavism (inverting results=slower)
- Atavism (progressive=slower)
- Sound (on=slower)

Output considerations

The following is a list of recommendations for yielding interesting patterns and evolution.

- Count-based rules almost guarantee interesting output. Logical rules tend to be more solid in their output, the change from one pattern to the next often marked.
- If you are using logical survival rules, specify many of them, or allow random rules to be generated. If only a few logical structures ensure survival of a cell, the patterns generated will be dead by and large.
- Allow mutation.
- Allow epidemics, and select the setting that randomly chooses the type.
- Allow atavism.

- Specify colours that are not all different and motley. Use 1-2 base colours, varying their shades instead.

- Use a path that has contiguous positions, i.e. the next pattern lies close to the previous, so that you can see the progression flowing across the screen, instead of a random skipping.

- See to it that the interval at which mutation, epidemics and atavism occur, is different for each. It is generally more interesting to see each operate on its own, than a sudden confluence of these events.

- Generally, a simple initial pattern is better than a complex one. A simple initial pattern also ensures that atavism, if allowed, yield noticeable results.