NAME

attraction − interactions of opposing forces

SYNOPSIS

attraction [−−display *host:display.screen*] [−−foreground *color*] [−−background *color*] [−−window] [−−root] [−−window−id *number*][−−mono] [−−install] [−−visual *visual*] [−−points *int*] [−−threshold *int*] [−−mode balls | lines | polygons | splines | filled-splines | tails] [−−size *int*] [−−segments *int*] [−−delay *usecs*] [−−color-shift *int*] [−−radius *int*] [−−vx *int*] [−−vy *int*] [−−glow] [−−noglow] [−−orbit] [−−viscosity *float*] [−−walls] [−−nowalls] [−−maxspeed] [−−nomaxspeed] [−−correct-bounce] [−−fast-bounce] [−−fps]

DESCRIPTION

The *attraction* program has several visually different modes of operation, all of which are based on the interactions of a set of control points which attract each other up to a certain distance, and then begin to repel each other. The attraction/repulsion is proportional to the distance between any two particles.

OPTIONS

attraction accepts the following options:

−−window
Draw on a newly-created window. This is the default.

−−root Draw on the root window.

−−window−id *number*
Draw on the specified window.

−−mono
If on a color display, pretend we're on a monochrome display.

−−install
Install a private colormap for the window.

−−visual *visual*
Specify which visual to use. Legal values are the name of a visual class, or the id number (decimal or hex) of a specific visual.

−−points integer
How many control points should be used, or 0 to select the number randomly. Default 0. Between 3 and 15 works best.

−−threshold integer
The distance (in pixels) from each particle at which the attractive force becomes repulsive. Default 100.

−−mode balls | lines | polygons | tails | splines | filled-splines
In *balls* mode (the default) the control points are drawn as filled circles. The larger the circle, the more massive the particle.

In *lines* mode, the control points are connected by straight lines; the effect is something like *qix*.

In *polygons* mode, the control points are connected by straight lines, and filled in. This is most interesting in color.

In *splines* mode, a closed spline is interpolated from the control points.

In *filled-splines* mode, the splines are filled in instead of being outlines. This is most interesting in color.

In *tails* mode, the path which each particle follows is indicated by a worm-like trail, whose length is controlled by the *segments* parameter.

−−size integer
The size of the balls in pixels, or 0, meaning to select the sizes randomly (the default.) If this is specified, then all balls will be the same size. This option has an effect in all modes, since the ‘‘size’’ of the balls controls their mass.

−−segments integer
If in *lines* or *polygons* mode, how many sets of line segments or polygons should be drawn. Default 500. This has no effect in *balls* mode. If *segments* is 0, then no segments will ever be erased (this is only useful in color.)

−−delay microseconds
How much of a delay should be introduced between steps of the animation. Default 10000, or about 0.01 seconds.

−−color-shift int
If on a color display, the color of the line segments or polygons will cycle through the color map. This specifies how many lines will be drawn before a new color is chosen. (When a small number of colors are available, increasing this value will yield smoother transitions.) Default 3. This has no effect in *balls* mode.

−−radius
The size in pixels of the circle on which the points are initially positioned. The default is slightly smaller than the size of the window.

−−glow This is consulted only in *balls* mode. If this is specified, then the saturation of the colors of the points will vary according to their current acceleration. This has the effect that the balls flare brighter when they are reacting to each other most strongly.

In *glow* mode, all of the balls will be drawn the same (random) color, modulo the saturation shifts. In non-glow mode, the balls will each be drawn in a random color that doesn’t change.

−−noglow
Don’t do ‘‘glowing.’’ This is the default.

−−vx pixels

−−vy pixels
Initial velocity of the balls. This has no effect in **−−orbit** mode.

−−orbit Make the initial force on each ball be tangential to the circle on which they are initially placed, with the right velocity to hold them in orbit about each other. After a while, roundoff errors will cause the orbit to decay.

−−vmult float
In orbit mode, the initial velocity of the balls is multiplied by this; a number less than 1 will make the balls pull closer together, and a larger number will make them move apart. The default is 0.9, meaning a slight inward pull.

−−viscosity float
This sets the viscosity of the hypothetical fluid through which the control points move; the default is 1, meaning no resistance. Values higher than 1 aren’t interesting; lower values cause less motion.

One interesting thing to try is

```
attraction -viscosity 0.8 -points 300 \
-size 10 -geometry =500x500
```

Give it a few seconds to settle down into a stable clump, and then move the drag the mouse through it to make "waves".

−−nowalls
This will cause the balls to continue on past the edge of the screen or window. They will still be kept track of and can come back.

−−walls This will cause the balls to bounce when they get to the edge of the screen or window. This is the default behavior.

−−maxspeed
Imposes a maximum speed (default). If a ball ends up going faster than this, it will be treated as though there were .9 viscosity until it is under the limit. This stops the balls from continually accelerating (which they have a tendency to do), but also causes balls moving very fast to tend to clump in the lower right corner.

−−nomaxspeed
If this is specified, no maximum speed is set for the balls.

−−fast-bounce
Uses the old, simple bouncing algorithm (default). This simply moves any ball that is out of bounds back to a wall and reverses its velocity. This works fine for most cases, but under some circumstances, the simplification can lead to annoying effects.

−−correct-bounce
Uses a more intelligent bouncing algorithm. This method actually reflects the balls off the walls until they are within bounds. This can be slow if balls are bouncing a whole lot, perhaps because of -nomaxspeed.

−−graphmode none | x | y | both | speed
For "x", "y", and "both", displays the given velocities of each ball as a bar graph in the same window as the balls. For "speed", displays the total speed of each ball. Default is "none".

−−fps Display the current frame rate and CPU load.

ENVIRONMENT

DISPLAY to get the default host and display number.

XENVIRONMENT
to get the name of a resource file that overrides the global resources stored in the RESOURCE_MANAGER property.

XSCREENSAVER_WINDOW
The window ID to use with *−−root*.

SEE ALSO

X(1), **xscreensaver**(1)

COPYRIGHT

AUTHOR

Jamie Zawinski <jwz@jwz.org>, 13-aug-92.

Viscosity support by Philip Edward Cutone, III.

Walls, speed limit options, new bouncing, graphs, and tail mode fix by Matthew Strait. 31 March 2001