Musophobia

Cem Sina Cetin*

1 Introduction

Graphic artists have a wide variety of applications to use for digital painting. Although each application has its own solution to enhance the user experience, most of them rely on the same standard feature; a single brush, which is completely dependent on user input for location. Although this is required for a fully controlled painting process, making small changes on this feature yields unpredictable results. My proposal for an alternate brush paradigm is using multiple brushes (as seen in the application "PD Particles"), which are not completely under control but rather moving within trajectories with random deviations, simultaneously. The trajectories are defined by controllable parameters and the user input. Since the rate of obedience to user input is dependant on the parameters, users can define the rate of deviation and thus switch between finger painting and generative painting, without changing the set of tools.

2 Exposition

Musophobia is a multi-touch painting application, where the user controls multiple brushes simultaneously. These brushes are in the form of floating particles with fixed rest locations. Particles are always visible on the canvas, but they do not paint unless the painting is selected. Particles always respond to user interaction, whether or not the painting mode is on. When the interaction ends, particles return to their rest locations.

The application runs in several states, which changes the interaction behavior of the particles. There are 5 modes of interaction, which are namely "blob", "spin", "orbit", "fall" and the default, rest state. The rest state shows the particles at their fixed locations. When the user touches the screen, nearby particles escape from the point of touch. In blob state, the particles cluster around the point of touch at a fixed location relative to the touch position. Cumulatively, particles act as a single brush in this mode. In spin state, particles also cluster around the point of touch, but they revolve around the origin (i.e. the touch location) at random distances and speeds. Orbit state causes the particles to travel on circles, around random centers, radii and speeds. Fall state dislodges particles from their current locations and let them fall in the direction of gravity. States of interaction work mutually exclusively. When two of these states are active, one of them overrides the other. In addition to these mutually exclusive interaction states, there are three separate visualizations. "Triad" visualization creates triangles by using 3 nearest points in the rest state, "Arrow" replaces particles with speed and direction sensitive arrowheads and "Hide" state makes the points invisible. The combinations of these two sets of interaction and visualization define the resulting behavior.

Each particle is attributed with a number of parameters. These parameters are randomly generated during the first initialization of the canvas and used for animation traits like speed during spin state or distance from center at blob state. New position of a point is calculated relative to the location on the previous frame, so that the continuity of the animation is not compromised. Since the motions are coherent in consecutive frames, the particles are guaranteed to paint continuous, fluent paths, no matter at what frequency the interaction states are changed. While the paths are continuous for each unique particle, the combined outcome forms harmonious patterns. Following image shows the path of 10 and 999 particle respectively, during transition from blob state to orbit state.

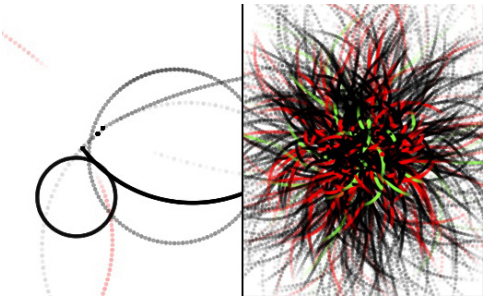

Figure 1: *Trajectories during blob to orbit transition*

Although the interaction variables are generated at initialization, the user is granted the ability to tweak these values to control the rate of random behaviour and basic appearance options. Change of appearance involves the particle amount, size and color options, where the user can either select predefined colors from pickers, or force particles to continuously change their hue (the rainbow mode). The remaining options, such as the spin radius, or blob radius, help the user to define the boundaries for the particles to appear within specific states. For instance, the lower the spin radius is, the denser the particles will be in a defined area, which, in other words, forces them to collapse into a smaller area and act more like a brush. In addition, the auto hide option makes the particles invisible if there are no touch events, during spin and blob modes. Using auto hide, forces the particles to behave like a regular finger painting; touching and dragging the finger makes the particles visible, while moving the finger away stops painting again.

3 Conclusion

Replacing the single brush painting with a system of particles with parameterized, random behaviors makes it possible to create generative images and finger painting within the same canvas, by using the same tools. Although such system makes it more difficult to practice classical painting methods, it offers efficient and easy ways to build up generative patterns to include in an artistic work.

Musophobia is developed for and currently available on Apple's iPhone and iPad. Musophobia has over 5,000 users globally and the promising results of this adaptation show that, with a larger touch screen device and thus a canvas, features of Musophobia will be easier to use and will make it possible to create more complex, more detailed works.

References

DAVE SHREINER, MASON WOO, J. N., AND DAVIS, T. 1995. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional.

RITCHIE, D., 2006. Pd particles, August. http://www.thebest3d.com/pdp/.

*e-mail: i.am@cemsinacetin.com

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007