

Chapter 12 Graphical User Interfaces

In this chapter we will consider the development of graphical user interfaces (GUIs) that can facilitate user interaction with a program. The J system provides laboratories and demos that can be used to learn about GUI programming. We make no attempt to duplicate those recommended resources, but try to offer some examples that are useful and interesting in their own right. We begin by exploring a palette editing form defined in the *fvj4/ped8.ijs* script that was developed to modify palettes used in chaotic attractor images to make them more aesthetic. In subsequent sections we will take a look into the workings of that script.

12.1 Experiment: Visual Form for Exploring Palettes.

We begin by loading a script that loads the palette editing form.

```
load '~addons/graphics/fvj4/ped8.ijs'
```

Press the open button and open the image file *nearp4m.bmp* from the *fvj4* addon directory. The result should be a form as shown in Figure 12.1.1. First experiment with the scrollbar buttons below

and to the right of the image. Next experiment with the magnification button to see more (or less) of the image at once. Notice that the action on selection drop down menu (near the bottom right) is on “blend”. Select the white palette entry in the upper left of the palette window by right clicking once. Now hold the shift key down and right click on an entry near the center of the palette. The palette should be modified to smoothly change along the interval defined by the clicks. Shift click on another colored entry and finally on the black entry, which should result in a grayscale palette and image. You can use the “palette history forward” and “palette history forward” buttons to quickly review those palettes.

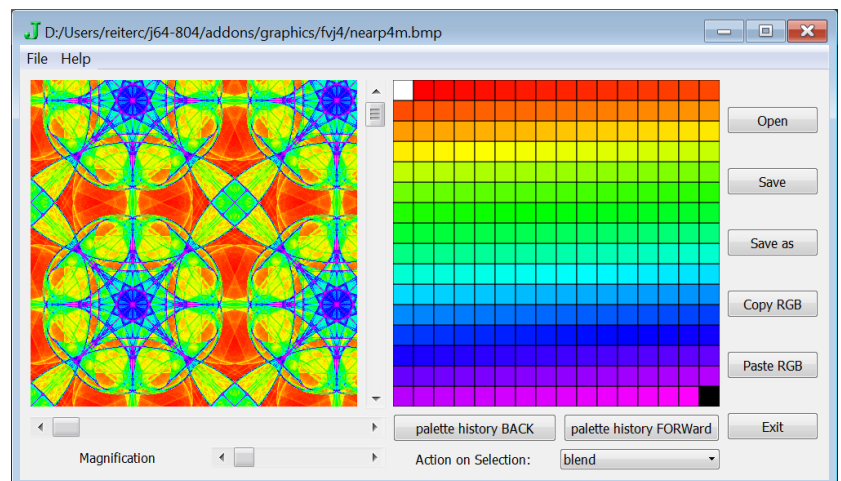


Figure 12.1.1 The Palette Editing Form

Next select “reverse” in the “Action on Selection” dropdown menu. Experiment with a few selections with that option. Next try “inverse” and then “random 4”. Note that if you make a selection of the second palette entry and then shift-right-click on the second to the last palette entry, and then after a pause, shift right click again, one can swiftly view many random palettes. If your reflexes are slow, and you want to save one that you already replaced, don’t worry, use the palette history buttons. Figure 12.1.2 shows such a random palette.

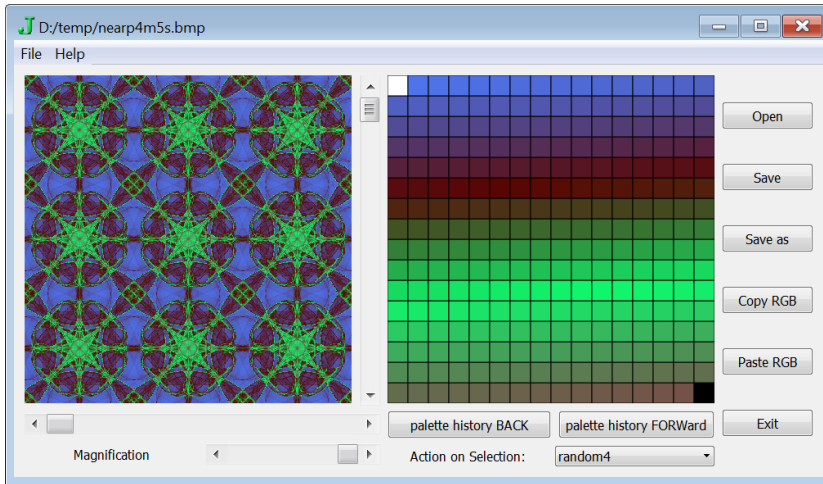


Figure 12.1.2 Form with a Random Palette

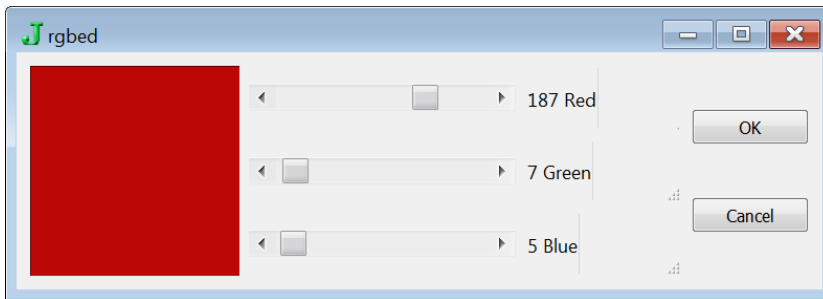


Figure 12.1.3 The RGB Editing Form

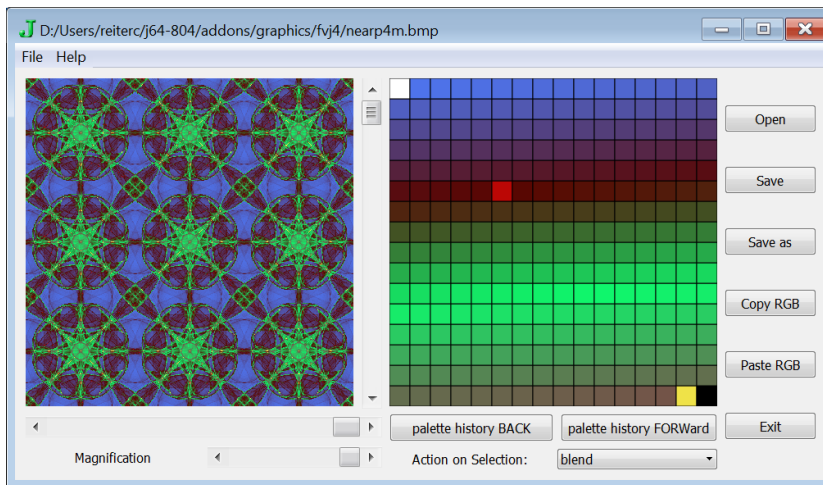


Figure 12.1.4 Form with Two Entries Changed

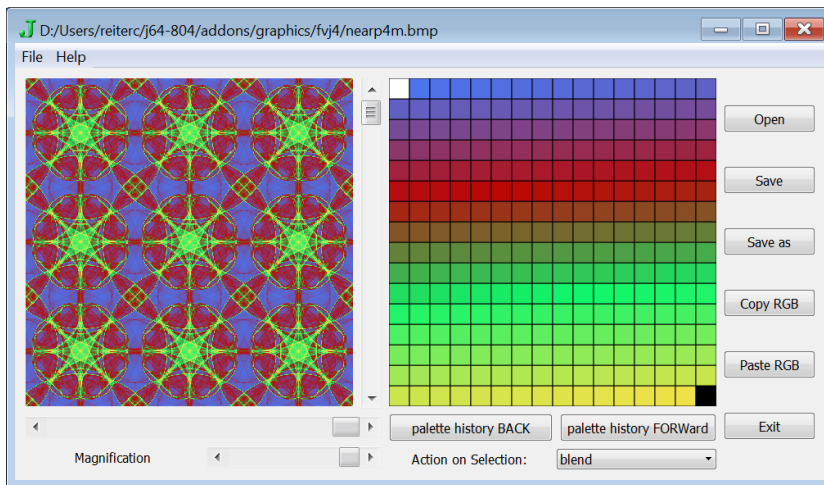


Figure 12.1.5 Changed Entries Blended into the Palette

We can also modify palette entries by double clicking a palette entry. Figure 12.1.3 shows the form that is used to change a particular palette entry. Figure 12.1.4 shows the image palette with two modifications: an entry in the dark range has its red intensity increased and a yellow entry has been added near the end. Run similar experiments with your own random palette. Figure 12.1.5 shows the result after the palette has been blended between the ends and the modified entries.

Next we turn to how to save chaotic attractor images as 8-bit paletted **.bmp* files so that you can edit the palette with the form described above. At the end of Section 9.3 there was a discussion of recreating images from the raw *VRA* data. We suppose there is a file *c:/temp/cd/hd.vra* and the associated **.png* is also in the same directory. We load scripts for their utilities dealing with chaotic attractor files and for writing paletted bitmaps. We can either reconstruct the array of indices into the palette using the **.vra* file, or, if we know the palette used, we can reconstruct it using the **.png* file.

```
load '~addons/graphics/fvj4/chaotica.ijs'
load '~addons/media/imagekit/pal_rw_bmp.ijs'
path=: 'c:/temp/cd/'
$VRA=: arrayread path, 'hd.vra'
1000 1000
$b=: P254 i. read_image path, 'hd.png'
1000 1000
b-:|.cln254 VRA
1
(P254;b) pal_write_bmp path, 'hd.bmp'
1001078
```

The **.bmp* file saved this way can be opened with the palette editor we are discussing in this section.

12.2 Visual Form for Selecting an RGB triple.

Graphical user interfaces provide a powerful means for users to interact with programs. The menus, scrollbars, mouse clicks and visual feedback offered by GUIs make them easy to use. Moreover, with careful design they can greatly facilitate user creativity and productivity. We will not offer an extensive introduction to the topic, but we will attempt to encourage, by example, readers to use those facilities in JQT when appropriate. The JQT demos give a rich resource for examples of the available features. In this section and the next two we look at some of the underlying code of GUI utilities that are part of the *fvj4* addon. In particular, here we look at parts of the code related to the form that allows an individual palette entry to be edited.