

## Creating Weavable Color Patterns, Part 2: A Design Application

Most handweavers design using drafts — threading sequences, treadling sequences, tie-ups, and the resulting drawdown. The warp and weft threads may be assigned colors, and the resulting pattern viewed in a “color draw-down”. This process assures a weavable pattern. There is no way to produce one that is not weavable.

In the previous article in this series, we described how to ensure weavability in algorithmically constructed patterns and showed transformations on weavable patterns that preserve weavability [1].

We have experimented with a different approach to constructing weavable color patterns; one in which a designer constructs color patterns “from scratch” but not in the context of drafting. Instead, the designer uses an interactive computer application that prevents anything that would result in an unweavable pattern.

This article describes that application and gives examples of its use.

### The Application

The application displays several windows: an interface window that provides controls for the user, a design window in which the user creates designs, and a palette window that displays the colors available for design.

#### The Interface

The application interface, shown in Figure 1, displays three colors associated with the left, middle, and right mouse buttons, respectively.

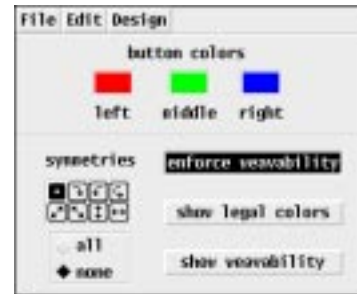


Figure 1. The Interface Window

The initial colors are red, green, and blue. These colors can be changed as described later.

#### The Design Window

The design window consists of a rectangular array of cells. Initially all cells are colored with the middle mouse button color. See Figure 2.

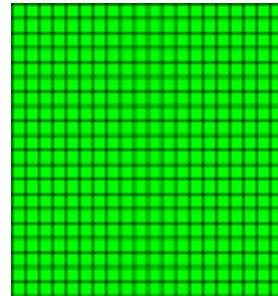


Figure 2. The Design Window

If the user clicks on a cell in the design window, the cell is colored with the color associated with the button used — provided the result would be weavable. If the result would not be weavable, the change is not made and there is an audible alert.

The user also can click and drag to color several cells at one time. The test for weavability is not made until the mouse button is released.

If the result would not be weavable, the application backtracks, removing the most recently colored cells until there is a weavable result.

### The Palette Window

The color associated with a mouse button can be changed by clicking with that button on a cell in the palette window. The initial palette provides a range of colors as shown in Figure 3, but other palettes are available and can be selected as described later.

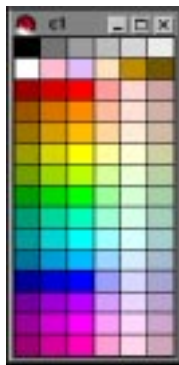


Figure 3. The Palette Window

### Symmetrical Designing

The application supports symmetrical designing in which cells in symmetric positions are colored. Symmetries can be selected from the symmetry panel on the application interface. See Figure 1.

The default is no symmetry, so only the color of the cell under the mouse pointer is changed. This is indicated by the highlighted button in the upper-left corner of the symmetry panel. Various combinations of symmetries can be selected by clicking on the icons for individual symmetries. See Reference 2 for a detailed explanation.

All symmetries can be enabled by choosing the all radio button below the symmetry panel. Figure 4 shows a design produced by using symmetries.

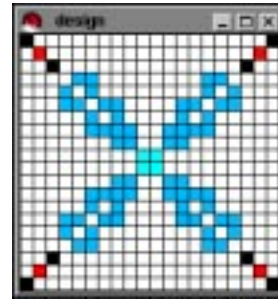


Figure 4. A Symmetrical Design

### Layout

The default layout for the design window is a 20 × 20 array of 10-pixel square cells. See Figure 5.



Figure 5. The Default Layout

The cells need not be square. Their widths and heights are determined by the values in sequences in the first two fields of the layout window. These sequences are repeated as necessary to fill out the specified number of cells. A scaling factor is applied to these values.

Finally, a palette can be specified. The default palette is c1, which is shown in Figure 3. See Reference 3 for information about the other palettes that are available.

The initial colors for buttons for a new design are colors in the given palette that are close to red, green, and blue, respectively.

Figure 6 shows a layout based on the

Fibonacci sequence, which is reflected to produce a symmetric result. The palette g8 provides eight equally spaced shades of gray. The resulting design window is shown in Figure 7 and the new interface and palette windows are shown in Figures 8 and 9.

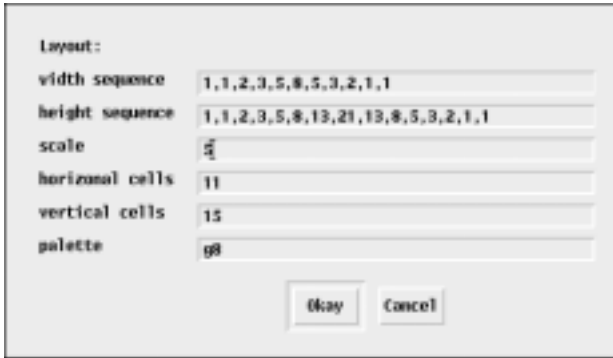


Figure 6. A Fibonacci Layout

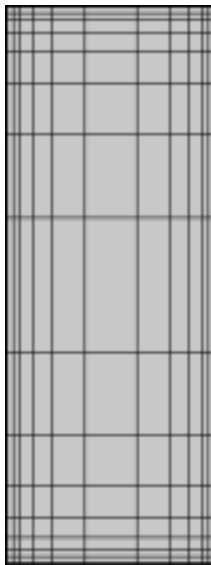


Figure 7. A Fibonacci Design Window



Figure 8. New Interface Window

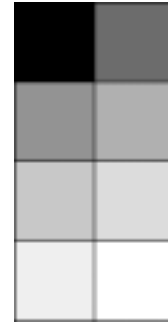


Figure 9. New Palette Window

### Enforcing Weavability

The enforce weavability button on the interface is a toggle, which initially is on. If it is off, changes in the design window are not checked for weavability.

One reason for not enforcing weavability is that it often is not possible to get from one weavable pattern to another by changing the colors of cells one at a time. One way to accomplish changes that preserve weavability but cannot be done piecewise is to disable weavability testing, make the changes, and then enable weavability testing. (Should the result not be weavable, the previous changes are undone as necessary the next time a change is made.)

### Legal Colors

The show legal colors button toggles the visibility of a window that mimics the design window. Clicking on a button color region on the interface window (see Figure 1) overlays on the legal color window all the cells in the design that could be made that color while preserving weavability. See Figure 10.

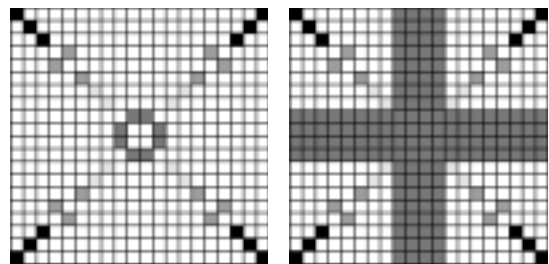
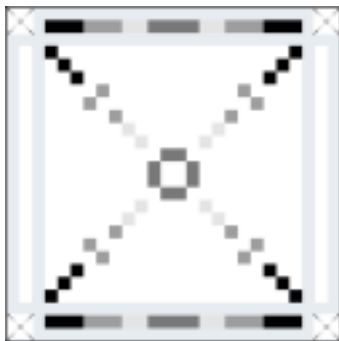


Figure 10. Design and Legal Colors Windows

The legal color window only shows individual cells that can be colored; it does not show all combinations of cells that together would preserve weavability. Indeed, all cells in the design window can be made one color with a result that is trivially weavable.

**Showing Weavability Testing**

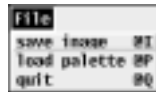
The show weavability button toggles the visibility of a window that shows the result of the last weavability test [4]. This result shows the row and column colors determined by the test. See Figure 11.



**Figure 11. Weavability Solution**

**Menus**

The File menu, shown in Figure 12, has items for saving an image of the design window, loading a custom palette database, and quitting the application.



**Figure 12. The File Menu**

The Edit menu, shown in Figure 13, provides items for undoing and redoing the last change to the design window.



**Figure 13. The Edit Menu**

There is no limit to the number of changes that can be undone or redone; it is possible to

move backward and forward through the entire history of a design. As indicated, the stack for saved designs can be cleared, which frees the memory they occupy.

The Design menu, shown in Figure 14, provides items related to the design.



**Figure 14. The Design Menu**

The new item brings up the layout dialog described previously. Entire designs can be saved and loaded as indicated. The clear item clears the design to a single color. It brings up a dialog in which the user can choose between the left, middle, and right button colors.

**Experience with the Application**

Our experience with the application is limited and the only users so far have been familiar with the concepts of weaving and color weavability but not experienced weavers.

If the user doesn't think in terms of row and color assignments (doesn't know the rules) but considers the application as a game, the results can be more frustrating than interesting.

Starting with a "blank" (solid-colored) design window, cells with a second color can be drawn in any fashion, since all two-colored designs are weavable. However, it may be impossible to change any cell to a third color unless the second color was used judiciously.

The user learns to reserve areas of cells so that they can be colored with other colors, the most notable being stripes (it's always possible to change all the cells in a row or column to a new color).

Figures 15-19 show examples of weavable color patterns created using the application described here.

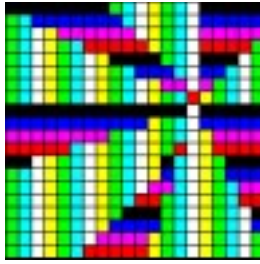


Figure 15. Vivid Geometry

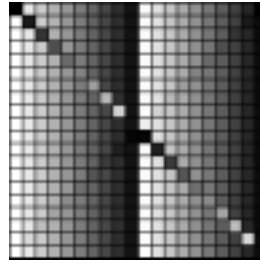


Figure 16. A Study in Grays

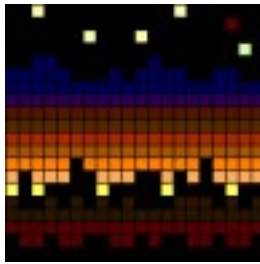


Figure 17. Skyline at Dawn

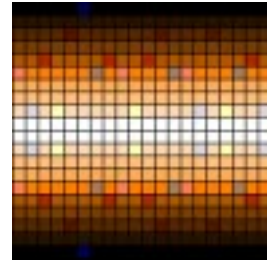


Figure 18. Shimmer

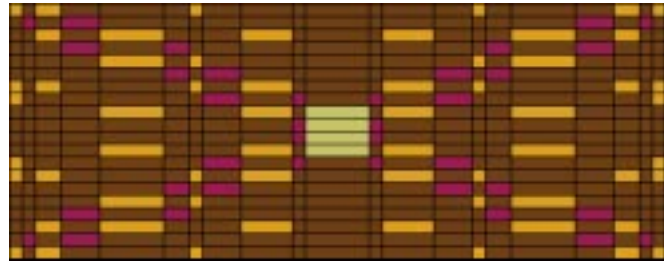


Figure 19. Parquet

## References

1. Ralph E. Griswold, "Drafting Weavable Color Patterns, Part 1", 2004:  
[http://www.cs.arizona.edu/patterns/weaving/webdocs/gre\\_col1.pdf](http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_col1.pdf)
2. Ralph E. Griswold *Penelope — A Pattern Tile Editor*, IPD234, Department of Computer Science, The University of Arizona, 1996:  
<http://www.cs.arizona.edu/icon/ftp/doc/ipd234.pdf>
3. Ralph E. Griswold, Clinton L. Jeffery, and Gregg M. Townsend, *Graphics Programming in Icon*, Peer-to-Peer, 1998.
4. Ralph E. Griswold, "Weavable Color Patterns":  
[http://www.cs.arizona.edu/patterns/weaving/webdocs/gre\\_cwev.pdf](http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_cwev.pdf)
5. Ralph E. Griswold and Madge T. Griswold, *The Icon Programming Language*, 3th ed., Peer-to-Peer, 1996:  
<http://www.cs.arizona.edu/icon/books.htm>

Ralph E. Griswold  
Department of Computer Science  
The University of Arizona  
Tucson, Arizona

© 2001, 2002, 2004 Ralph E. Griswold