

# DragPanels

Stephan J.C. Eggermont, Sensus

September 28, 2015

*DragPanels demonstrates how to make widgets that support click, double-click, drag-and-drop and context menus in Morphic. This is shown with a color panel and an avatar panel. Morphic is a powerful graphics environment, used in Self, Squeak, Cuis and Pharo.*

## 1 A color and avatar panel

---

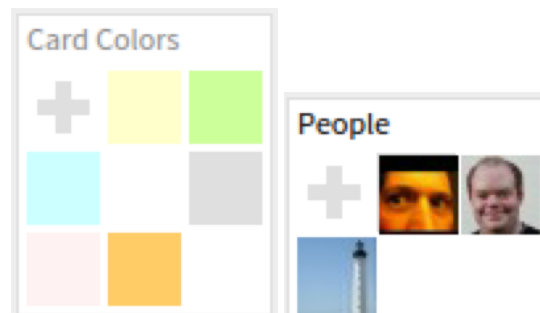


Figure 1: The panels

The first iteration (Figure 1) shows two panel windows that stay in front of other windows. They can be moved around by dragging the title bar part. The plus symbol can be clicked to add new elements to the panel. Each element is a square of 30\*30 pixels. An element can start a drag action where a copy of the element is dragged to a Morph that wants to receive it. The default action for dragging a color is setting the color of the Morph, for dragging an avatar it is setting the email.

## 2 A Drag Well

---

A DragWell is a small square widget. The default size is 30 px. It is derived from a BorderedMorph as a border might be shown.

```
BorderedMorph subclass: #DragWell
  instanceVariableNames: 'panel over dropSelector clickSelector'
  classVariableNames: ''
  category: 'DragPanels'
```

The default border can be white and 1 px

```

DragWell>>defaultBorderColor
  ^Color white

DragWell>>defaultBorderWidth
  ^1

DragWell>>defaultBounds
  ^0@0 corner: 30@30

```

### 3 Copying a Morph

---

A DragWell keeps a reference to the DragPanel it belongs to. This is stored in the panel instance variable. When Morphs are copied, they are by default copied very deep. We expect all nested submorphs to be copied too. For some instance variables that is unwanted. A link to a parent object is typically something that should be either shallow copied, so the reference remains to the same parent object, or it should be nilled or set to some default value. To make sure that that is what happens when copying a DragWell, there are two methods that potentially need to be created: `veryDeepInner:` and `veryDeepFixupWith:`. Here, we just want a shallow copy of all instance variables

```

DragWell>>veryDeepInner: deepCopier
  super veryDeepInner: deepCopier.

```

### 4 Hovering

---

The DragWell responds to mouse actions. To show that it does so, we want to give feedback when the mouse is over the DragWell.

```

DragWell>>mouseenter: evt
  "Handle a mouseEnter event, meaning the mouse
    just entered my bounds with no button pressed."

  super mouseEnter: evt.
  self over: true

DragWell>>mouseleave: evt
  "Handle a mouseLeave event, meaning the mouse
    just left my bounds with no button pressed."

  super mouseLeave: evt.
  self over: false

```

### 5 Loading the code

---

The code can be found on [www.smalltalkhub.com](http://www.smalltalkhub.com), in the repository StephanEggermont/DragPanels. Open the Monticello Browser. Add a new repository of type smalltalkhub.com. The owner is StephanEggermont, the project is DragPanels. User and password are only needed when you want to commit changes to the repository. Open the repository and load the latest version of DragPanels.