PostBoard - a prototype GroupStorm application

Application by Hans Andersen and Jeff Raymakers
Document by Hans Andersen

PostBoard is a cross-platform Group-Storm application for fluid content creation, organization, and annotation, intended for use with a large, low- to medium-resolution touch-screen display such as a SmartBoard.

PostBoard's interface and features mimic paper sticky notes placed on a whiteboard. Content can be drawn on the blank board, and this content can be copied off the board onto a virtual sticky note. The sticky notes reside "above" the board, and can be moved around and modified. The intent is that the whiteboard be used for scratch work and content creation, from which coherent pieces of data are extracted and formed and encapsulated by a note.

What does it mean to be a "cross-platform Group-Storm application for fluid content creation, organization, and annotation, intended for use with a large, low- to medium-resolution touch-screen display"? The following is a breakdown of the buzzwords:

Cross-Platform:
PostBoard is written entirely in Java2. A Java 1.3 version exists for running under Mac OS X, while a very slightly modified Java 1.4 version exists for running under Windows 2000 and ME. It should be runnable under any OS with a sufficiently recent Java VM.

GroupStorm Application:
PostBoard was written with iRoom technologies in mind. It uses the Event Heap 2 API to accept content from outside sources. Any application that can generate JPEG or GIF data and communicate with EHeap2 can asynchronously add content to PostBoard. Together with ImageSender, a small Java utility, virtually any existing application on virtually any computer can send images to PostBoard, simply by saving its output to a special directory. Users with laptops can add web content to the PostBoard, while users with illustration applications and one-touch scanners can quickly create and add pictures and sketches, all while other users actively manipulate content on the PostBoard.

Fluid:
PostBoard's interface is heavily influenced by PostBrainstorm, and shares many of its characteristics. The interface is as uniform and non-modal as possible, with no buttons, widgets, or visible menus of any kind. The default interaction is to draw, as if on a whiteboard. "Sticky Notes" automatically resize themselves to accommodate new pen strokes, without any explicit user intervention. All other actions and features are initiated via a FlowMenu interface. Furthermore, PostBoard uses a ZoomScape to reduce clutter and sort information.

Content Creation:
The primary means of interacting with PostBoard is through drawing. Thus, any information that can be drawn on a blank whiteboard, can be drawn into PostBoard.

Content Organization:
Any content drawn or imported into PostBoard can be copied or "lifted" off the whiteboard-like background onto a moveable "Sticky Note", which can then be rearranged, modified, and "stamped" or "merged" back onto the PostBoard's whiteboard-like surface. These virtual sticky notes serve to encapsulate indivudual thoughts, tasks, and pieces of information, and to distinguish them from the whiteboard-like background, which serves as a space for scratch work.

Content Annotation:
Images imported into PostBoard via the Event Heap appear as the background upon a new "Sticky Note". New markings and annotations can be added on top of the note, and erased again without disturbing the background. An annotated image can then be stamped or merged back onto the background for further scratch work.

Large Touch-Screen Display:
PostBoard was designed with touch-screens in mind, particularly large screens such as the SmartBoards in the iRoom. These screens generally run at comparatively low resolutions, such as 1024x768. ("low" in this case is relative to the iRoom Mural, which runs at ~4000x3000 resolution.) These resolutions are sufficient for most whiteboard-style uses, but they limit the size and resolution of imported image content. However, PostBoard can be viewed on any screen whose computer can accept a two-button mouse. It has been used successfully with large desktop displays (such as the Apple Cinema Display) and Wacom Tablets, although the tablet's indirect drawing can never be quite as intuitive as direct stylus-to-screen input.

FEATURES

Drawing

Drawing content is a simple matter of touching the screen and “finger-painting”. The stroke will be painted onto whatever sticky note it touches, or if none, then onto the background. Sticky notes automatically grow to accommodate strokes that start within their bounds.

The flowmenu is invoked with a “right mouse down”. On a system with a stylus, there is often a button mounted on the barrel of the stylus which modifies presses accordingly.

The color of the pen can be changed using the flowmenu, starting with a “down” gesture. A quick “down-up” gesture serves to set the color of the pen back to black.

The flowmenu is also used to access various eraser tools: a freehand eraser, a box-region eraser, and “erase all”. The latter completely erases the surface to which it is applied - either the whiteboard, or a single sticky note.

Creating Notes

Sticky notes are created by invoking the flow menu on the whiteboard surface, starting with an “up” gesture. From here, there are three options: “Copy Box”, “Cut Box”, and “Empty Box”. When one of these options are chosen, the user traces out a box enclosing the content that he wishes to be included in the note. The size of the note is the rectangle enclosing the point at which the flowmenu was invoked, and the point at which the single gesture ended. “Empty Box” makes an empty note the size of the selected region. “Copy Box” duplicates the whiteboard content that lies under the selected region, and puts that content into the new note. “Cut Box” acts as “Copy Box”, but subsequently erases the whiteboard under the selected region.

The fastest way to make a new note using whiteboard content with a two-part gesture starting with the flowmenu: An “up-down” gesture followed by a diagonal sweep across the area to be enclosed.

Notes can also be made by importing content. Outside sources can construct an event of type “PostBoardEvent”, and place in it a URL for a picture, with the key “PostPicture”, and place that event on the EventHeap. PostBoard will receive that event and fetch the picture at the corresponding URL, and create a new note that includes the picture as its “background”.

If a note has a picture in its background, than any erasing actions will erase any drawing that is on top of the picture, without disturbing the picture. In this way, an imported picture can be “wiped clean” of annotations.

Moving Notes

The flowmenu reveals different features and options when invoked on the whiteboard than it does when invoked on a sticky note.

The same “up-down” gesture that triggers the creation of a new note on the whiteboard, is used to “pick up” and move a postit. This is identical to the moving action in PostBrainstorm.

As in PostBrainstorm, two horizontal likes mark the start and end of the Zoomscape. A note that is placed such that its center lies above the upper line shrinks to a fraction of its native

size, while a note placed so that its center lies between the two lines shrinks part way. The zoomscape only applies to sticky notes, and not to whiteboard drawing. If a sticky note is created out of a whiteboard drawing that is located in or above the zoomscape, then the newly created note will immediately subsequently shrink down. If the note is moved out of the zoomscape, then it will return to its original size.

Other Note Operations

A note may be "stamped", which copies the note's content onto the whiteboard below it. A note may be "merged", which copies the note's content onto the whiteboard below it, and subsequently deletes the note.

A note may have one of a variety of pastel shades, much like their paper counterparts. A note may also be rendered semitransparent through the use of a quick "up-right, down-left" gesture.

Problems with PostBoard

No saving or loading - PostBoard stores its content as bitmaps, which are difficult to write to disk in Java. In Java 1.4, this is less of an issue, and we are working on implementing saving and loading in that version.

No undo or redo - PostBoard's paint-oriented implementation makes the infinite undo very difficult to implement, and any implementation would consume large amounts of disk space.

No rescaling of images, other than the zoomscape - Our FlowMenu implementation doesn't yet support "circular actions" like PostBrainstorm does, so we have no interface for doing so.

Non-rectangular pieces of content can waste lots of screen real estate - Guilty as charged. We're planning on implementing non-rectangular selections and groupings in our next project.

Sketches scaled in the zoomscape become unintelligible - Another drawback to our paint model. If we had used a vector-based draw model, then this wouldn't be a problem.

No text support - PostBoard is an early attempt at making a GroupStorm application for brainstorming, and support thus far has been focused on images. PostBoard doesn't support text, nor will it ever support text; however, future projects certainly will!

PostBoard occasionally pauses for several seconds, sometimes in mid-action - We've seen this with the 1.4 version on some Windows 2000 machines, and we suspect that it has to do with Java's garbage collection.