API for iScribble, a touch interface using JAVA

Kevin Tang, Adarsh Kowdle, Dhruv Batra

Quick Start

To start, call the init() method from SquiggleFrame. What's returned is a handle on the frame of the entire GUI. With this SquiggleFrame, you can get the Squiggle, which holds most of the information, by calling getSquiggle(). With the SquiggleFrame and Squiggle, you have access to all the methods I have written.

To create a java string array in Matlab:

A = {'winter.jpg', 'sunset.jpg', '200', '150'}

Integer arrays are just like normal Matlab arrays.

An example initialization call:

```
Import SquiggleFrame;
frame = javaMethod('init', 'SquiggleFrame', A)
squiggle = frame.getSquiggle()
```

SquiggleFrame Class

Void getSquiggle()

-returns the Squiggle object that is under the SquiggleFrame

Void setMode(Integer)

-takes an integer as input, and sets the mode to the corresponding integer.

-also used to repaint current window (by setting to the same mode)

MODES:

0 -> all the images laid out as tiles for selection

1 -> all panels available, including segmentations and confusion maps

2 -> full screen view of one image for scribbling

Integer getMode()

-returns an integer that represents the current mode

Static SquiggleFrame init(String[])

-initial main function used to start everything, returns the SquiggleFrame for the entire GUI.

-called with string array that represents the initial images that can be selected at mode 0. the last two elements of the array must be the width and height in pixels(as strings) of these thumbnail images when first displayed in mode 0.

Squiggle Class

String getImagen()
-returns the string name of the current image.
-all names are converted to lower case.

Void setImage(String)
-sets the current image to the filename pointed to by the parameter string.

Void setImageSize(Integer w, Integer h)
-sets the image size of the main image to be in a panel of width w and height h

Integer[] getImageSize()
-returns an integer array containing (width, height) of main image's actual resized size (with aspect ratio kept)

Void setMaxPixels(Integer)
-sets the maximum number of pixels the user is allowed to scribble to the parameter integer

Integer getMaxPixels()
-returns the current maximum number of pixels the user is allowed to scribble

Integer getImagePixels()
-returns the current number of pixels that have been scribbled on the current image

Integer getTotalPixels()
-returns the total number of pixels that have been scribbled on all images so far

Void setScore(Integer)
-sets the score of the user to the parameter integer

Integer getScore()
-returns the current score of the user

Void setActualImages(String[])
-sets the set of actual images (original images that are in the top selection pane) to the files denoted by the filenames in the String array. Thus, the String array is just an

array of filenames.

String[] getActualImages()
-returns the array of strings that are the filenames of the set of actual images

Void setSegmentations(String[])
-same as setActualImages but sets the set of segmented images

String[] getSegmentations()
-returns the array of strings that are the filenames of the set of segmented images

Void setConfusionMaps(String[])
-same as setActualImages but sets the set of confusion map images

String[] getConfusionMaps()
-returns the array of strings that are the filenames of the set of confusion map images

Void setMapWH(Integer w, Integer h)
-sets the size of the small thumbnail versions of the actual/segmented/confusion map images in the right panel to pixel width w and pixel height h.

Integer[] getMapWH()
-gets as an integer array the current sizes of the thumbnail versions of the actual/segmented/confusion map images.

Void setTileWH(Integer w, Integer h)
-sets the size of the small thumbnail versions of the actual images from mode 0 to pixel width w and pixel height h.

Integer[] getTileWH()
-gets as an integer array the current sizes of the thumbnail versions of the actual images in mode 0.

Void setButtonWH(Integer w, Integer h)
-sets the size of the buttons in the top panel to width w and height h.

Integer[] getButtonWH()
-gets as an integer array the current sizes of the buttons in the top panel.

Void setDone(Boolean)
-sets the current value of done to the given Boolean.
-done is set to true whenever the done button is pressed, representing that the user is done scribbling.

Boolean getDone()
-gets the current value of done, true/false.

Boolean getNextGroup()
-gets the current value of nextGroup, true/false.
-nextGroup starts as false, and when the user clicks on the nextGroup button to go to the next group of images, nextGroup is set to true.

Void setNextGroup(Boolean)
-sets the value of nextGroup to the parameter boolean.

Vector<Vector<Integer[]>> getSquiggle()
-gets the current squiggles in the buffer as a vector of vector of integers arrays, where each integer array is stored as [x, y].
-empties the buffer of current squiggles after the get
-also, the color of the squiggle (0 for Red for Background, 1 for Orange for Foreground) is stored as the very last integer array in each squiggle as [0 or 1, -1]

Vector<Vector<Integer[]>> getSquiggle(String)
-gets all the squiggles for the file of name given by String

Vector<Vector<Integer[]>> getAllSquiggles()
-returns all of the squiggles that have been scribbled up until now

Vector<String> getAllSquigglesNames()
-returns the file names of all the squiggles that have been scribbled up until now

Void associate(String s1, String s2, String s3)
-associates filenames together as a data structure, where s1 is the actual image file name, s2 is the segmentation file name, and s3 is the confusion map file name.