Loading

Creation

Storage

1. Canvas Operations

A canvas is the primary storage type in EZ-ASCII. All of the image modification happens on this type. Internally, it is represented as a two-dimensional array of integers referred to as intensities. This canvas can be loaded from an existing image file or it can be created manually.

* 1. Intensity Mapping and Granularity

The intensity mapping contains a table of intensities to characters. The is done to simplify a programmers interaction with the canvas. A programmer should not have to consider

which exact character they are drawing but instead just how intense each point should be. This makes drawing much simpler. Each intensity mapping also has an associated *granularity* as defined later in this section. The default mapping will include all printable ASCII characters ordered by the amount of pixels they take up in each character space in ascending order.

A custom mapping can be defined in the following way:

MAP <- {I0:C0, I1:C1, … ,IN:CN}

*MAP* is a keyword referring to the intensity map. Each *I* is an intensity and the corresponding *C* is the character mapped to that intensity. Any reference to the intensity mapping will reference to the most recent assignment of *MAP* or the default if none have been assigned.

With respect to an intensity mapping the granularity defines how many intensities should be used. If the granularity and the size of the intensity mapping are the same, then all intensities will be used. Otherwise, the intensities will be spread out as close to possible as evenly amongst all of the values in the intensity map. The granularity must always be at least 2 and the first and last intensity value will correspond to the first and last values in the intensity mapping. For intensities 1 to n-1 (where n is the number of intensities), the difference between each is defined by:

diff = (n - 2) / ((g-2) + 1)

Where n is the size of the intensity mapping and g is the granularity specified.

* 1. Printing

Using the -> operator, a canvas can be saved to a file or printed to a console output. The effect of this operator is to apply the intensity mapping to the canvas and print the corresponding canvas to console or a file.

Saving to a file can be done using the following syntax:

canvas -> file\_path, render

The *canvas* must be a valid identifier of a canvas object as described by section *Y.Z. render* must be a boolean. If render is true, the intensity mapping will be applied to the canvas before printing and the end result will be a file of characters. If render is false, the canvas will be printed as is. If render is false, the filename will be suffixed with “.i” to represent an EZ-ASCII intensity file. The file name must be a path to a valid file location in the form of string type. If the file does not exist, it will be created and the contents of the canvas, mapped into the appropriate characters will be printed to it. If the file already exists, it will be overridden.

Printing to the console can be done in a similar way:

canvas -> out, render

Again, the *canvas* must be a valid identifier of a canvas object and render must be a valid boolean value. o*ut* refers to the out keyword described in section Y.Z. In this case, the contents of the map will be printed to standard output instead of a File.

* 1. Creation

The are two ways to create a canvas in EZ-ASCII. One is to take an existing image and load it. The other is to create a blank canvas and use it.

To load an image the syntax is as follows:

canvas <- file\_path, granularity

The canvas must be a valid identifier for a canvas object as described by section Y.Z. file\_path must be a valid string indicating the full or relative path to the file. The granularity must be an integer indicating the granularity to be used with the current intensity map to convert the image. If the loaded image is an EZ-ASCII intensity file indicated by the suffix “.i” as described in the previous section, it will be loaded as is. Otherwise, the image will be converted to black and white and then Floyd-Steinberg dithering will be applied to the image in order to generate intensities to draw. These intensities will be normalized using the given granularity and loaded to a new canvas.

To load an image with a blank canvas

canvas <- blank(width, height, granularity)

The c*anvas* must be a valid identifier for a canvas object as described by section Y.Z. The width and height must both be valid integers describing the dimensions of the canvas to be created. The initial intensity of each character in the canvas will be 0. The granularity must be a valid integer describing the granularity associated with the canvas.