Functions:

A user will be able to create functions that will act as a blocks of code that can be called when desired. They will accept a list of input parameters and return some value at the end of their execution. It will be possible to specify no input parameters or return nothing if this is desired. Recursive functions will be allowed. There are two types of function calls: fun and vfun. fun means for returning some value but vfun means returning nothing in the end of function calls. Also there are some default functions in the language.

blank ([width], [height], [granularity]):

-blank - A function that returns a black canvas of some specified size.

- Params: -- width - The width of the canvas in character spaces (Int)

--height - The height of the canvas in character spaces (Int)

--granularity - The initial granularity of the canvas (Int)

load([filename], [granularity]):

-load - A function that loads an image into a canvas from a specified file.

- Params: -- filepath - The file path to the image in question. (String)

--granularity - The granularity at which the image should be loaded. The higher this is the more "detailed" the picture will be. (Int)

ie.: can <- load('pic.jpg', 10);

ci[argument]:

- ci is a canvas identifier will return a canvas of the same dimensions as ci which contains only the desired elements. The rest are blank.

- Argument: Different argument represent different ways of selecting things: the following is possible: - Select 1 point : ci[x, y]

- Select range of points:

--ci[x1:x2, y1:y2] - Rectangle (x2 and y2 must be greater than x1 and y1 respectively)

--ci[x, y1:y2] - Vertical Slice

--ci[x1:x2, y] - Horizontal Slice

-- ci[,y] or ci[x,] or ci[,] -Shorthand for full vertical slice, full horizontal slice, or all

- Select points based on boolean expression:

--ci[exp] - Get all of the elements with intensity conforming to the boolean condion. boolean expression must be in format ([cond][0-255][(&&)|(||)])\* where condition is one of >, <, =, <=, >=

Scope:

A declared object can be visible only within a particular function. Also a declaration is not visible to declarations that came before it. A variable name cannot be referred before declared.

Includes:

To include one file in your source at compilation time, say

#include "filename"

This is useful for putting a lot of heavily used data definitions and #define statements at the beginning of a file to be compiled. As with #define, the first line of a file containing a #include has to begin with a `#'. And #include can't be nested -- an included file can't contain another #include.