COMS W4115 PLT

Spring 2014

Dainis Kiusals (dvk2102)

ImageVis (IV) Language Reference Manual

Lexical Conventions

Comments

Comments in IV will be either contained between the /* and */ character sequences, or will begin with a // character sequence and terminate with the newline character.

Whitespace

Whitespace comprises of one or more space, tab or newline characters.

Keywords

The following keywords are reserved for IV:

for	while	if	else
print()	int	float	String
getWidth()	getHeight()	getRed()	getGreen()
getBlue()	getName()	getSize()	setName()
setRed()	setGreen()	setBlue()	create()
displayOnPanel()	incrRGB()	PolarToCartY()	PolarToCartX()
incrTexture()	getTexture()	setBW()	getBW()
setTexture()	drawPoint()	stringOnPanel()	drawLine()
incrDistance()	getDistance()	setDistance()	getRGB()

Identifiers

Identifiers in IV are composed of upper and lowercase letters and numbers and the '_' character. IV is a case sensitive language and identifiers need to be case sensitive as well. The first character must be a letter (either case) or '_'.

Constants

Numeric constants are decimal numbers which may be followed by a fractional part (after a decimal). These constants may be of type int or float (if they contain a fractional part). Examples include 3, 1.2, 135.267, 3.37 and 2.0.

String constants are declared with a String type variable or used in an expression with the string constant contained within double quotes (example: "something").

Variables

Local variables are declared within a block, and only have local scope and lifetime. Global variables are declared at the top of the program and have global scope and indefinite lifetime. Declaration is written as the type followed by the variable name or constant with assignment of a value.

Types / Objects

Numeric - numbers may be of type int for integers or float for floating point numbers. They may be defined as constants as described above.

String - string type object represent values such as image names and sizes, and may be displayed on a visualization panel with the stringOnPanel() function.

Image - represents a 2D image (which may be read in from a file or created within the program. Images support the create(), displayOnPanel(), getWidth(), getHeight(), getRed(), getGreen(), getBlue(), setRed(), setGreen(), setBlue, getName(), setName() and getSize() functions.

Array - 1D arrays may be declared for lists of strings, integers or floats with '[]' braces used at declaration with the size (number of elements) contained within the braces. Individual elements are referenced by using an integer to reference the elements through a 0-based index. 1D arrays may also be declared for Image objects.

RGBHistograms - histograms for each image with red, green and blue histogram components for each Image. The functions for the RGBHistogram object includes create(), getRGB() and incrRGB()

TEXHistograms - histograms for each image with a texture component for each image. The functions for the TEXHistograms object includes create(), incrTexture and getTexture().

BWValues - 2D array of integer black & white values for each pixel of a corresponding Image, used to compute the texture values for the image. Functions supported by this object are create(), setBW() and getBW().

TextureValues - 3D integer array of texture values for each pixel for multiple images, used for image pair texture comparison. Functions supported by this object include create(), setTexture() and getTexture().

DistanceMatrix - 2D array of float values representing the color or texture distance between each pair of images in the image set. The distances between an image and itself correspond to the (0,0), (1,1),…etc. entries in the matrix. Functions supported by the object are: create(), getDistance(), incrDistance() and setDistance().

Panel - object representing 2D image visualization to present to the user. Functions for the panel object include create(), drawLine() and drawPoint()

Operators

+ addition between two numeric values or concatenation of two String objects

- subtraction between two numeric values, negation of a unary numeric variable

* multiplication of two numeric values

/ division of two numeric values

= assignment operator

+=, -= addition to, subtraction from current value

== equality operator

!= inequality operation

>=, >, <=, < Greater than or equal, greater than, less than or equal and less than operators

Punctuation

; - marks the end of a statement, separator for 'for' loop initiation, condition and iteration

. - prefix for object functions

() - prefix and suffix for parameter list

, - separator for parameters in a parameter list

{} – denotes the beginning and end for a block of code, defining the boundaries of local scope

Conditionals

if – the if statement is written with an expression which is either true or false (ex: if (x == 3)). After the expression either a single statement ending with a semicolon or a code block delimited by {} follows and is executed if the expression is true.

else – the else statement must follow the single statement or code block which follows an if statement. It also has a single statement or code block which follows which is executed if the corresponding if expression is false.

for – the for loop statement is written with a parameter list of 3 elements – an initial value, condition and iterator, which define how many times the following statement or code block will be executed. The condition and iterator is evaluate / executed once per loop iteration (ex: for(i=0; i<10;i++).

while – the while loop statement is written with a single Boolean expression which is evaluated before each iteration of the loop, followed by a single statement or code block which is executed if the expression is true (ex: while(x < 4)).

Functions

print(String) – print a String to the console

create() – create an Image object. If a String filename is specified in the parameter list it will be copied from the image file. If 2 integers are in the parameter list they will specify the size of a blank, new Image object. Create can also be used to create an RGBHistogram or TEXHistogram, with one integer parameter specifying the number of Image objects in the histogram. Create can also be used to create BWValues objects with 2 integers specifying the width and height, or a TextureValues object with 3 integers specifying the number of images, width and height. It can also create a DistanceMatrix with 1 integer for the number of Images, and Panel object with 2 integer parameters for the width and height in pixels of the panel.

getWidth(Image) – returns the integer value of the width of the Image object in pixels

getHeight(Image) – returns the integer value of the height of the Image object in pixels

getRed(int, int) – returns the integer red component (0-255) of the specified X, Y pixel in the Image

getGreen(int, int) – returns the integer green component (0-255) of the specified X, Y pixel in the Image

getBlue(int, int) – returns the integer blue component (0-255) of the specified X, Y pixel in the Image getRGB(int, int, int, int) – returns the integer red component (0-255) of the specified RGB in the Image (first parameter) in the RGBHistogram

setRed(int, int) – sets the integer red component (0-255) of the specified X, Y pixel in the Image.

setGreen(int, int) – sets the integer green component (0-255) of the specified X, Y pixel in the Image.

setBlue(int, int) – sets the integer blue component (0-255) of the specified X, Y pixel in the Image.

getName() – returns the String name of an Image

setName(String) – set the name of an Image object created in the program

getSize() – returns the integer size in number of pixels for an Image

displayOnPanel(int, int) – displays an Image object on the panel at the specified X, Y location for the upper left corner of the image

incrRGB(int, int, int, int, int) – increment the count by an integer value (fifth parameter) for the Image identified by the first integer in the parameter list which corresponds to the index for an Image in the RGBHistogram object by the RGB values of the second through fourth parameters.

getTexture(int, int, int) – get the texture for a TEXHistogram if two integer identifying the corresponding Image and texture bucket in the Histogram are presented, or texture for a TEXValues object with three integers specifying the image and X,Y coordinate

incrTexture(int, int) – increment the texture value for the texture value for an image specified by the first integer parameter and texture histogram bucket identified by the second integer

setTexture(int, int, int) – sets the texture value for an image specified by the first integer parameter in a TEXValues object at the X,Y coordinate specified by the last 2 integers in the parameter list

getBW(int, int) – get the integer black & white value at X,Y position in the BWValues object

setBW(int, int) – set the integer black & white value at X,Y position in the BWValues object

drawPoint(int, int) – draws a point at the X,Y position on the Panel object

drawLine(int, int, int, int) – draws a line from X1,Y1 to X2, Y2 on the Panel onject

stringOnPanel(String, int, int) – draw a String on the Panel object starting at position X,Y getDistance(int, int) – returns the float distance between two Images in the DistanceMatrix object specified by indices Image1 and Image2

setDistance(int, int) – sets the float distance between two Images in the DistanceMatrix object specified by indices Image1 and Image2

incrDistance(int, int) – increments the float distance between two Images in the DistanceMatrix object specified by indices Image1 and Image2

PolarToCartX(int, int) – conversion of integer radius and degree value to Cartesian X coordinate

PolarToCartY(int, int) – conversion of integer radius and degree value to Cartesian Y coordinate