ISE 108 Function Quick Reference

This document briefly summarizes some common Processing commands and system variables that you may need for your programs.

System Variables

`key` — Indicates what key was last pressed on the keyboard

`mouseX` — Indicates the current *x* (horizontal) coordinate of the mouse pointer

`mouseY` — Indicates the current *y* (vertical) coordinate of the mouse pointer

Processing Methods

Drawing

`size ( width, height )` — Sets the canvas (window) size. Add `P3D` as a third argument to draw in 3D

`background ( <color> )` — Redraws the window background in the supplied color

`fill ( <color> )` — Changes the inside color of any new shapes

`stroke ( <color> )` — Changes the outline color of any new shapes

`smooth ()` — Turns on anti-aliasing, so curved lines appear smooth (not jagged)

point (x, y) — Draws a point at coordinates (x, y)

`line ( x1, y1, x2, y2 )` — Draws a straight line from (x1, y1) to (x2, y2)

`rectMode()/ellipseMode()` — Used to set the drawing mode for rectangles and ellipses to `CORNER` mode or `CENTER` mode. In `CORNER` mode, the x and y coordinates indicate the top-left corner of the box in which the shape is drawn. In `CENTER` mode, the shape is drawn in a box that is centered around the x and y coordinates.

`rect ( x, y, width, height )` — Draws a *width* x *height* rectangle at the supplied *x* and *y* coordinates, according to the drawing mode (see above). Defaults to CORNER mode.

`ellipse ( x, y, width, height )` — Draws an ellipse inside a *width* x *height* rectangle (bounding box) at the supplied *x* and *y* coordinates, according to the drawing mode (see above). Defaults to CENTER mode.

`triangle (x1, y1, x2, y2, x3, y3)` — Draws a triangle with its vertices at (x1, y1), (x2, y2), and (x3, y3).

User Interaction

void mousePressed () — This function is called automatically when the mouse button is pushed down.

void mouseClicked () — This function is called automatically when the mouse button is pushed down and then released.

void keyPressed () — This function is called automatically when a key is pressed on the keyboard.

JoptionPane.showInputDialog (text) — Display a dialog box that displays the specified message and returns the user input as a String. Requires you to add

import javax.swing.*;

at the top of your code.

Integer.parseInt (text) — Attempts to convert the input text (a String) into an integer value

Double.parseDouble(text) — Attempts to convert the input String into a double value

Text

PFont — variable type that holds a font/typeface for text displayed in the program window

createFont (typeface name, size in points, true/false) — returns a PFont object created from the specified typeface at the specified point size (72 points = 1 inch tall). The third argument is true or false based on whether the font characters should be antialiased (smoothed).

loadFont (filename) — loads a .vlw font file and returns it as a new PFont object

PFont.list() — returns an array of Strings that lists all the typefaces that are installed on the current computer system

textFont (PFont) — tells Processing to use the specified PFont for all further text rendering

text (data, x, y) — Draws the specified text (a String) in the program window at coordinates (x, y)

String methods

Note: all of these methods must be called on a specific String variable, e.g., myString.length()

length() — returns an integer that holds the total number of characters in a String

charAt (position) — returns a char variable representing the character at the specified position. Character positions begin at 0, and run through (length - 1)

indexOf (text) — Returns the starting index of the first occurrence of the specified text in a String. If the text is not found, this method returns -1.

replace (old, new) — Returns a new String where every occurrence of the substring *old* has been replaced by the substring *new*.

substring (start) — Returns a new String containing all characters from position start up through the end of the source String

substring (start, end) — Returns a new String containing all the characters from position start up to (but not including) position end from the source String

trim() — Returns a new String containing all the characters from the source String except for any leading or trailing whitespace (spaces between characters are left in place).

equals (String) — Used to compare two Strings. Returns true (a boolean) is the Strings have identical contents, and false otherwise.

compareTo (String) — Returns 0 if the two Strings are identical, a negative value if the source String is smaller (alphabetically before) the argument, and a positive value otherwise. "abc".compareTo("xyz") returns a negative value; "xyz".compareTo("abc") returns a positive value.

Images

PImage — variable type that can store the data for an image

loadImage (<filename>) — Loads the image file named <filename> (which is a String) and returns a new PImage object

image (image, x, y) — Display the PImage object image in the program window, with its top left corner at coordinates (x, y)

image(image, x, y, width, height) — Display the PImage object image in the program window, with its top left corner at coordinates (x, y), stretched to width pixels wide and height pixels tall

tint (<color>) — tints all images drawn after this statement with the specified color and/or transparency value

loadPixels () — Call this function on a PImage object to fill its internal pixels[] array for modification, e.g., myPic.loadPixels(). This pixels[] array consists of values of type color (which takes grayscale or RGB values as its arguments).

updatePixels() — Apply any changes made to a PImage's pixels[] array, so that they will be shown the next time the PImage is displayed.

height — a PImage field/constant that gives the total height of the image, in pixels

width — a PImage field/constant that gives the total width of the image, in pixels

red(<color>) — returns a floatcorresponding to the red component of this pixel.

green(<color>) — returns a float corresponding to the green component of this pixel.

blue(<color>) — returns a float corresponding to the blue component of this pixel.

Translation and Rotation

translate (x, y) — Shift the origin (0, 0) from the top left corner by the specified x and y amounts. Add a third integer argument to shift the origin back and forth along the Z axis

rotate (angle) — Rotate the canvas clockwise around the origin (around the Z axis) by the specified angle (a value in radians)

radians (degrees) — Convert the specified number of degrees into radians for rotate()

rotateX (angle) — Rotate the canvas along the X axis by the specified angle (in radians)

rotateY (angle) — Rotate the canvas along the Y axis by the specified angle (in radians)

box (width, height, depth) — Draw a 3D box centered around the origin with the specified dimensions

sphere (radius) — Draw a sphere centered around the origin with the specified radius

pushMatrix () — Save the current translation and rotation settings for later use/restoration

popMatrix () — restore the last saved translation and rotation settings to the drawing canvas

Miscellaneous Functions

`random ( <value> )` — Returns a random float between 0 and (*value* - 1)

`int (<value>)` — Converts <value> to an integer

`println ( <string> )` — Prints <string> to the console/screen

frameRate (<value>) — Slows down the frame rate (how frequently draw() is called) to value frames per second. Defaults to 60 frames per second.