

LCC 6310

The Computer as an Expressive Medium

Lecture 11

Overview

Programming concepts

Text & typography

Processing, pp. 737-740

Assignment 3 questions?

Project 2 - examples & discussion

PFont

PFont is the Processing class for manipulating fonts

Like PImage for images

Use PFont with these methods

PFont loadFont() – loads a font

PFont createFont() – converts a font to the correct format from a font name that's installed on the computer, or from a .ttf or .otf file inside the sketches "data" folder

textFont(PFont font, int size) – sets the current font once it's loaded

text(String s, int x, int y) and **text(String s, float x, float y)** – draws a string in the current font at the given x, y location

Simple example

```
// the fonts must be located in the data directory
size(200,200);
PFont metaBold = loadFont("Meta-Bold.vlw.gz");
PFont univers66 = loadFont("Univers66.vlw.gz");
textFont(metaBold, 44);
text("word", 10, 40);
textFont(univers66, 44);
text("word", 10, 80);
```

Let's try it in [Processing](#)...

Use fill() to change the color of text

```
// the fonts must be located in the data directory
size(200,200);
PFont metaBold = loadFont("Meta-Bold.vlw.gz");
PFont univers66 = loadFont("Univers66.vlw.gz");
fill(0, 255, 0);
textFont(metaBold, 44);
text("word", 10, 40);
textFont(univers66, 44);
fill(255, 0, 0);
text("word", 10, 80);
```

Let's try it in [Processing...](#)

textAlign sets the alignment

textAlign(LEFT)

x, y is the upper left hand corner of the text

textAlign(RIGHT)

x, y is the upper right hand corner of the text

textAlign(CENTER)

x, y is the upper center of the text

```
PFont metaBold = loadFont("Meta-Bold.vlw.gz");
textFont(metaBold, 20);
textAlign(RIGHT); text("word", 100, 50);
textAlign(CENTER); text("word", 100, 100);
textAlign(LEFT); text("word", 100, 150);
```

Let's try it in [Processing...](#)

Producing text effects (the fun stuff)

All the transform methods apply to drawing text

That means you can translate, rotate, and scale text

Combined with the draw() loop, this means you can move text around the screen in real time

Remember, the movement of the rocket and asteroids in our asteroids example was just translation and rotation

So you can easily make textual machines where text moves around the screen!

Farscape example: rotating text in 3D

Rotating an image one way and the text the other

```
...
// position the text
pushMatrix();
translate(250, 200, 0);

// rotate text clockwise around the y axis
rotateY(-radians(angle));
text(farscape, -twidh/2, 12);
popMatrix();

angle++;
...

```

Let's look at this example in [Processing...](#)

Processing example: typing

But first, let's talk briefly about special key handling...

We've seen how to use `keyPressed` for letters or spaces, e.g.

```
void keyPressed() {
  if (key == 'a' || key == 'A') {
    // do something
  }
}
```

But what if we want to know if the user has pressed a special key, e.g. backspace or escape?

We'll need to check the ASCII table for the correct ASCII code, i.e. a numerical representation of characters

<http://www.asciitable.com/>

ASCII codes for some frequently used special keys:

8 = backspace	10 = line feed	27 = escape
9 = tab	13 = carriage return	127 = delete

Key events with ASCII codes

```
void keyPressed() {
  char k;
  k = (char)key;
  switch (k) {
    case 8:
      println("Backspace"); break;
    case 127:
      println("Delete"); break;
  }
}
```

Let's try this in [Processing](#)...

Note: you can use the ASCII code to check for letters as well

(Just check the ASCII table for the values, 65-90 = A-Z, 97-122 = a-z)

And now: Processing typing example

Let's take a look at a nice typing example:

<http://www.processing.org/learning/examples/typing.html>

And play with it in [Processing](#)...

Assignment 3

Posted online, due **Friday October 5**

- A3-01: Create a subclass of `PImage` that implements a `mosaic(int blockSize)` method. The `blockSize` parameter specifies how big the mosaic block is (e.g. `blockSize = 4` would mean the mosaic block size is 4 pixels by 4 pixels). The `mosaic` method should replace each block of pixels in the image (e.g. if `blockSize = 4`, each block of 4 by 4 pixels) with the average color value of the pixels in that block. Look at the `Pixelate->Mosaic` filter in photoshop for an example of what this image operation does. Demonstrate your new class by drawing an image with several different block sizes.
- A3-02: Write a small app that demonstrates kinetic text. Your app should allow the user to type something and move the text around in some way while they type. For example, the user might type text on a line, but slowly the words or letters start drifting apart, or perhaps the line starts bending, or the words and letters flutter to the bottom of the screen, etc. Of course you shouldn't exactly copy any of the typographic examples in `Processing` or that you find on the web (though using such examples for inspiration, as a place to start modifying code, etc. is fine).

Project 2 discussion

Let's look at some of your project 2s...

Volunteers? (This is a good way to get some feedback from the whole class... if not I'll pick some...)

Things to think about

Concept: how does the drawing tool emphasize algorithmic generation/modification/manipulation? does it provide an overall theme or concept for the types of pieces it allows users to create? is it coherently done and easily grasped? is it original?

Presentation: does the drawing tool provide an appealing interface for user interaction? what are the criteria of visual appeal for a tool?

Interaction: is it clear? affordances? responsiveness? mappings? constraints? how does the tool balance constraints with freedom of expression for the artist?

Taking apart the source code

Volunteers? (If not I'll pick some...)

Remember...

For **Thursday** this week: Theory Readings

Two presenters (you know who you are!)

Everyone else: prepare one discussion question for each reading

Augmenting Human Intellect: A Conceptual Framework - Engelbart (NMR pp.93-108)

Computer Lib/Dream Machines - Nelson (NMR pp.301-339)