

Lecture VII:

Images + libraries

Mark Huiskes

LIACS, Leiden University

Introduction to Programming, Media Technology MSc



Lecture Schedule

15 Sep	Lecture I	Variables, data types, operators + assignments
Today	Lecture II	Functions + mouse interaction
29 Sep		no class
6 Oct	Lecture III	Conditions + loops
13 Oct	Lecture IV	Arrays
20 Oct	Lecture V	Classes + designing complex programs
27 Oct	Lecture VI	Recursion, data structures + sorting
3 Nov		no class
10 Nov	Lecture VII	Images and libraries
17 Nov	Lecture VIII	Max/MSP/Jitter (Edwin van der Heide) with 3 lab sessions!
24 Nov		Question session

Today

1. Tips + Tricks
2. Assignments previous lecture
3. Images
4. Libraries

Tips + Tricks

Processing JS: <http://processingjs.org/>

also open-source;

uses javascript, so can run without Java plugin

Recursion

- Recursion is a method of solving a problem by reducing it to a simpler problem of the same type
- Recursive function: a function that calls itself
- Make sure that you don't get an infinite sequence of function calls

Common Pattern

Make a problem-solving function:

- Make some (well-chosen) step(s) toward the solution
- You are left with a smaller problem similar to the original problem: call the problem-solving function again with arguments for the smaller problem
- When the problem is small enough to provide the answer directly, you return this answer (instead of calling the problem-solving function again)

Assignment VI.1

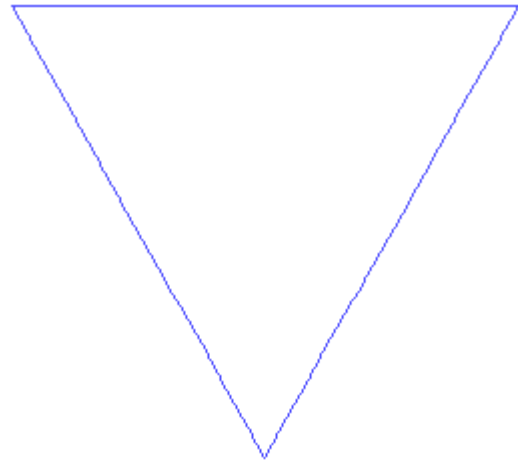
Assignment VI.2

Wikipedia

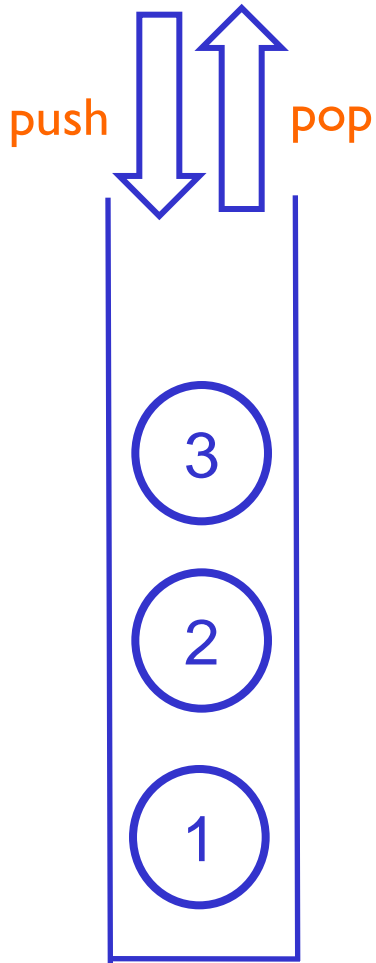
A fractal is generally "a rough or fragmented **geometric shape** that can be split into parts, each of which is (at least approximately) a reduced-size copy of the whole," [1] a property called **self-similarity**.

Mathworld

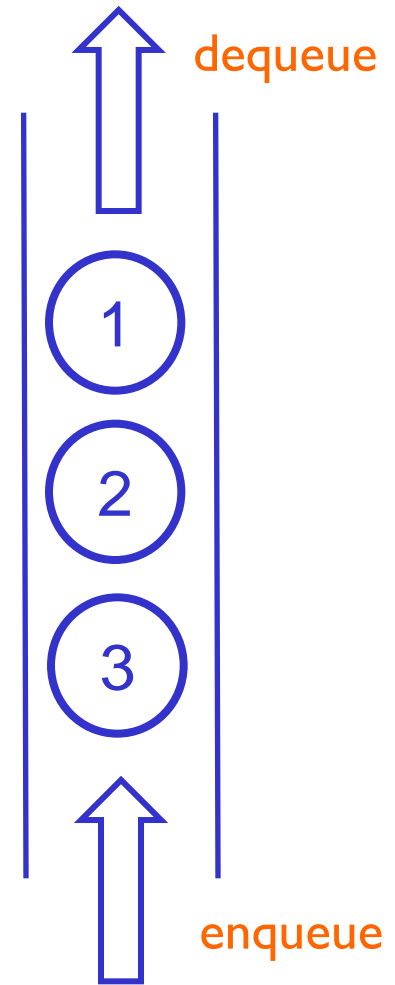
A fractal is an object or quantity that displays self-similarity, in a somewhat technical sense, on all scales. The object need not exhibit exactly the same structure at all scales, but the same "type" of structures must appear on all scales.



Data Structures



Stack: LIFO



Queue: FIFO

Assignment VI.3

Sorted List

Sorted List

- In our implementation: list of integers stored as a sorted array
- Interface methods:
 - `void insert(int value)`
 - `int remove(int position)`
 - `int length()`
- Position 0 corresponds to the smallest item
(and position `length() - 1` to the largest one)

Example VI.6

Bubble sort

example:

9, 5, 100, 17, 3

Images

- Main data type for images: **PImage**

```
PImage im;  
im = loadImage("test.jpg");
```

```
(or im = createImage(66, 66, RGB);)
```

properties & methods: see PImage reference

- Displaying an image on the applet canvas

```
image(im, x, y);
```

or:

```
image(im, x, y, width, height);
```

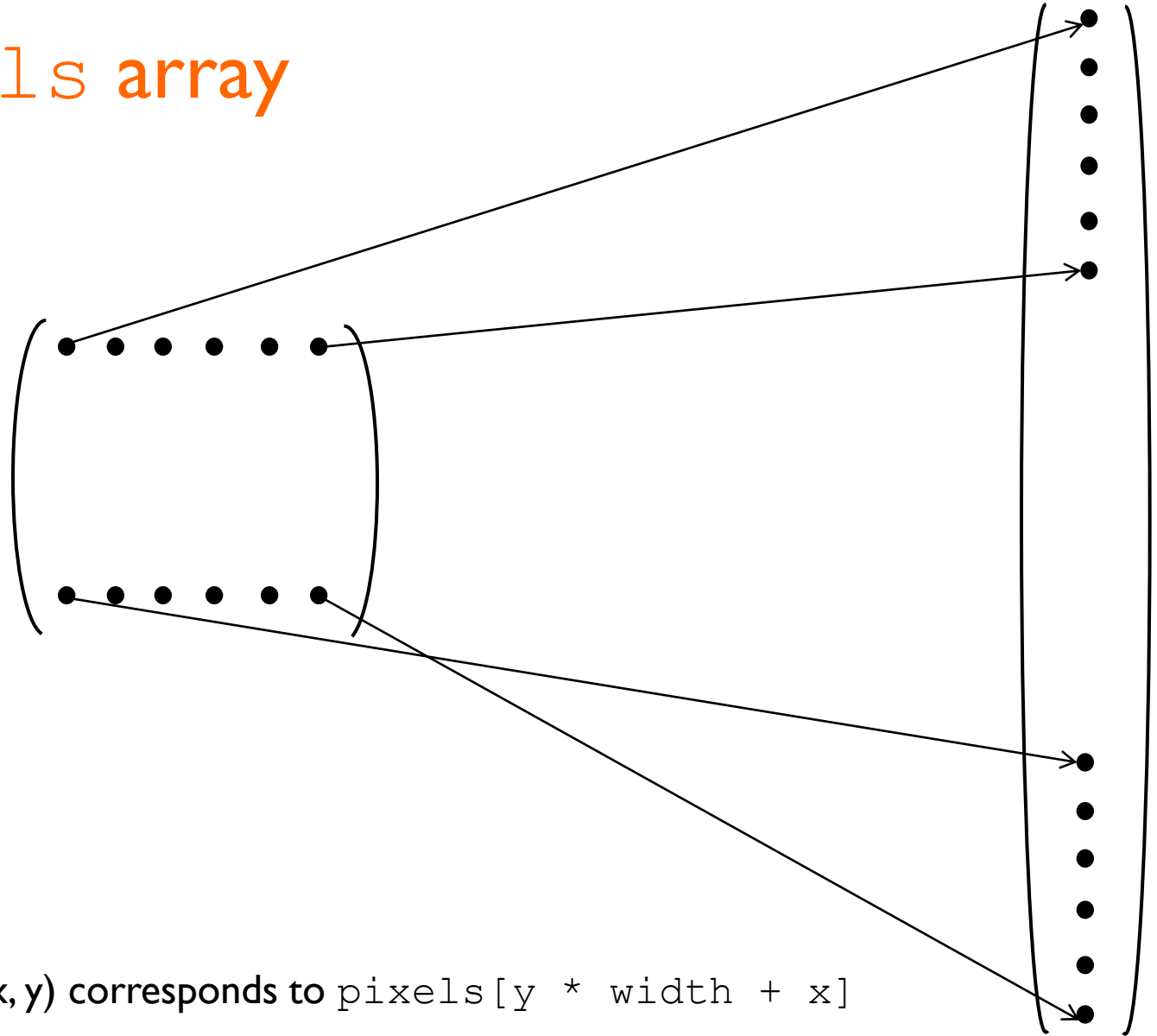
Pixel access

- first call the `loadPixels()` method:
- pixels are then available in the `pixels[]` property
- finish by calling the `updatePixels()` method

Example:

```
PImage im = loadImage("tower.jpg");
int n_pixels = (im.width * im.height);
im.loadPixels();
for (int i=0; i < n_pixels; i += 2) {
    im.pixels[i] = color(0, 0, 0);
}
im.updatePixels();
image(im, 0, 0);
```

The pixels array



- pixel at coordinate (x, y) corresponds to `pixels[y * width + x]`
- `pixels` is of type `color[]` ; `color` is built-in type to store a pixel color value

Example VII.1

if `loadPixels()` is called as a regular function it fills a global `pixels[]` array with the color values of the pixels on the current canvas

Example VII.2

Libraries

Libraries

- Main mechanism for **code re-use**
- In Processing/Java libraries contain classes
- Quality of libraries important characteristic of language
- Processing:
 - **Core** libraries- already in distribution
 - **Contributed** libraries- by community: download and put in libraries directory
 - Add import statement: by hand or through menu (Sketch/Import Library)

Example VII.3

Traer physics

<http://www.cs.princeton.edu/~traer/physics/>

Example VII.4

JMyron Video

<http://webcamxtra.sourceforge.net/>

Lecture Schedule

15 Sep	Lecture I	Variables, data types, operators + assignments
Today	Lecture II	Functions + mouse interaction
29 Sep		no class
6 Oct	Lecture III	Conditions + loops
13 Oct	Lecture IV	Arrays
20 Oct	Lecture V	Classes + designing complex programs
27 Oct	Lecture VI	Recursion, data structures + sorting
3 Nov		no class
10 Nov	Lecture VII	Images and libraries
17 Nov	Lecture VIII	Max/MSP/Jitter (Edwin van der Heide) with 3 lab sessions!
24 Nov		Question session

Exam

Tuesday, December 1

10:30 – 13:30

Snellius, room 413

and remember the question session
(see Lecture Schedule at beginning)

IP exam 04-12-2007; 17 students

