Assignment 1.3

Posted January 27
Before next class

Overview

This assignment is intended to explore functions, parametric forms, and object oriented programming in a design context.

1. Code flow

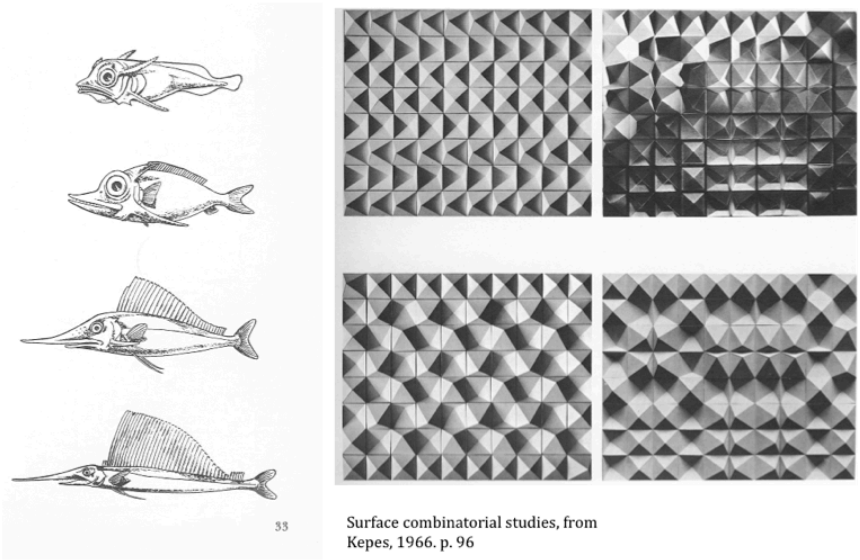
Consider the following code. First, try to “be” the computer and write down the outcome in a piece of paper. Then run the code and compare its outcome with what you wrote. Was the result what you were expecting?

```
void setup(){
 println("a");
 functionA();
 println("b");
}

void draw(){
  println("c");
  functionB();
  println("d");
  functionA();
  noLoop();
}

void functionA(){
  println("e");
  println("f");
}

void functionB(){
  println("g");
  functionA();
  println("h");
}
```

Document briefly the results in your blog.

Part 2. Modularity and parametric form

Take one of your previous sketches; modularize it by creating functions for drawing the element and defining its behavior. Think about what the element's parameters are, and what kind of "design space" they configure. Navigate that design space using a loop and iteratively calling the function with different parameterizations. Make it architectural if you wish.



Surface combinatorial studies, from Kepes, 1966. p. 96

Part 3. A Dynamic System—Object Oriented Design

Take the code from the previous point and re-arrange it to be Object Oriented (i.e. write a class for your object. Remember to include a) data, b) constructor, and c) functions). See the lecture notes for reference. Use a loop to instantiate many objects of the class you defined. Add extra functions to your object and explore the potential of OOP to create a dynamic system.

Part 4. Libraries

This part of the assignment asks you to learn how to use libraries in Processing. Libraries are organized packages of code that provide for special functions (generally) tied to a particular purpose. Look at the following libraries and read about what each of them does. When you have understood what their different purposes are, choose one and use it to expand the functionality of one of your previous programs.

- ControlP5 (http://www.sojamo.de/libraries/controlP5/)
- Toxic Libs (http://toxiclibs.org/)
- Unfolding (http://unfoldingmaps.org/)
- Sudden Motion (http://shiffman.net/p5/sms/)
- Leap Motion (https://github.com/heuermh/leap-motion-processing)