Aesthetic Programming

Class 08- Code and Objects

Mini exercise6 discussion- A Generative Program:

Noise/Perlin noise:

Joachim: https://github.com/djhest/Mini_ex/tree/gh-pages/Mini_ex6

Kristine: https://github.com/kris03/AP-17/tree/master/mini_ex6

Game of life:

Benjamin:
https://github.com/Epsilon99/AestheticProgrammingCourse/tree/gh-pages/MiniExcercises/MiniEx_6
Magnus:
https://github.com/MagnusJMJ/APME/tree/master/miniex6

“Complex systems often exhibit chaos. Chaotic dynamics are non-linear and difficult to predict over time, even though the systems themselves are deterministic and follow a strict sequence of cause and effect. The non-linearity of chaotic systems results in the amplification of small differences, and this is what makes them increasingly difficult to predict over time. This process is usually referred to as sensitivity to initial conditions.”

(Galanter 2016, p. 163)

Mini exercise6 discussion- A Generative Program:

1) Discuss how your code is constructed
2) Discuss your work in relation complex system

Peer tutoring - p5.dom library

- How to import and use p5.dom library in your sketch?
- Provide sample code
- How may we use different HTML5 objects in p5.dom to think about the aesthetics of objects?

Objects

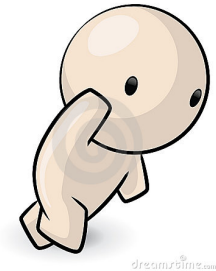
Object-oriented programming

- A program execution is regarded as a physical model, simulating the behavior of either a real or imaginary part of the world (Madsen et al, 1993, p.16)
- Objects are…used for representing (or modeling) physical phenomena (Madsen et al, 1993, p.18)
  - Simulate activities from the real world (Back 2013, p. 18)
- Objects are computerized material (Madsen et al, 1993, p. 18)

Objects

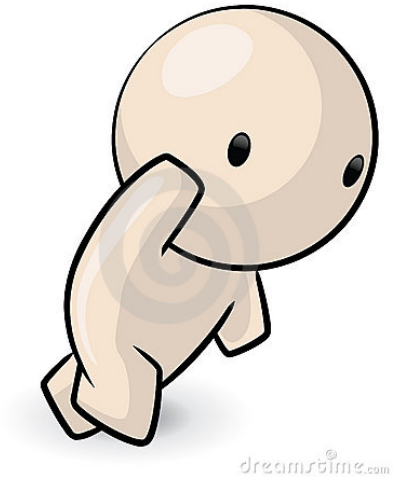
- Object with **properties, state and behaviors**
  - Object can do stuff -> functions
  - A class defines a group of methods (functions) and fields (variables)

- e.g A person with properties like brown hair, wear glasses, 165cm tall, female…
- e.g A person runs from Aarhus C to Aarhus N (run is the function)



Person
Hair color, with glasses? Height, gender, fromLocation, ToLocation
Run()

Objects



These describe what is a human being: (CLASS)

- **Weight**
- **Height**
- **Gender**
- **FromLocation**
- **ToLocation**

Instance object

Name: Peter Weight: 100 lbs Height: 170cm Gender: M From: Magazi To: Aros	Name: Susan Weight: 80 lbs Height: 150cm Gender: F From: Aarhus H To: Vibj

Objects

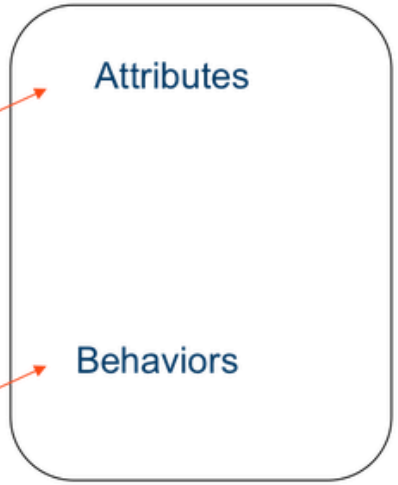
- Consider a car as an object. What are the attributes? What are the possible functions (behaviors)?

Object/Class

Class

- A blueprint for objects of a particular type
- Defines the structure (number, types) of the attributes
- Defines available behaviors of its objects

Object



http://205.173.41.8/staff/peck_george/notemplate/lessons/Ch03-ObjectsAndClasses/Ch03.html

Defining objects

1/ Car data/attributes:
- Car color
- Car s size
- Car x location
- Car y location
- Car z speed

2/ Setup:
- Initialize car color and size
- Initialize car location to starting - point
- Initialize the speed

3/ Draw/Behaviors:
- Fill background
- Display car at location with color and size
- Increment car's location by speed -> MOVE

A class specifies the structure, possible attributes and behaviours.
"A class is a factory that describes how to make an object, and describes the implementation and behavior of those embryonic objects. " (Back 2013, pp. 18-19)

```
function Car(getcolor, speed, xpos, ypos, size) {
        this.getcolor = getcolor;
        this.speed = speed;
        this.xpos = xpos;
        this.ypos = ypos;
        this.size = size;

        this.display = function() {
                stroke(0);
                fill(this.getcolor);
                rect(this.xpos,this.ypos,this.size,this.size);
        }

        this.drive = function() {
                this.xpos = this.xpos + this.speed;
                if (this.xpos > width) {
                        this.xpos = 0;
                }

        }

}
```

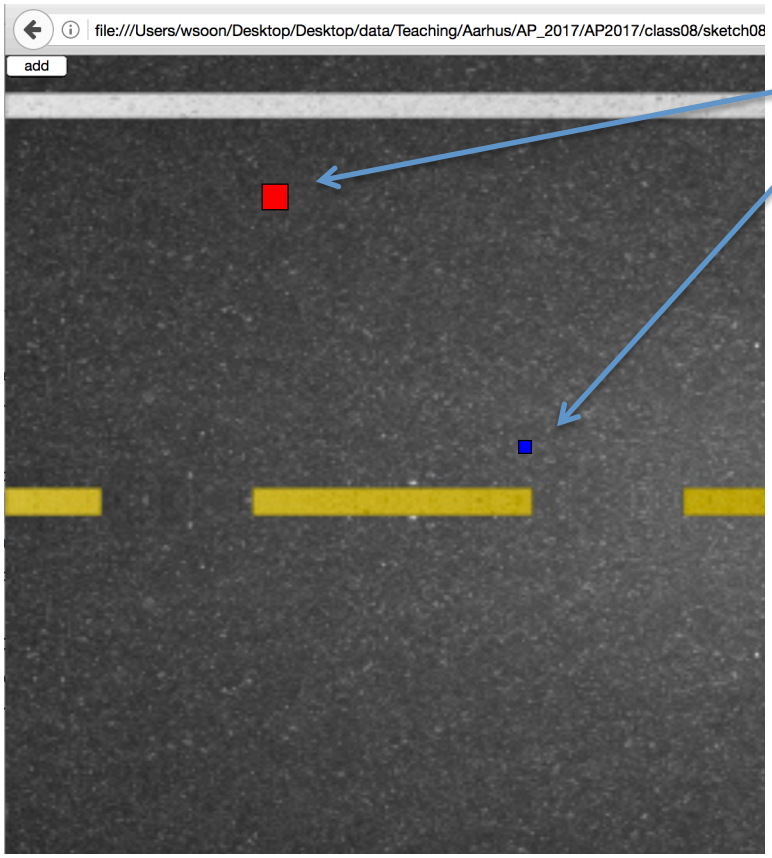
attributes

behaviours

behaviours

https://cdn.rawgit.com/AUAP/AP2017/18dfb50f/class08/sketch08/index.html

An object….



1) Holds specific values of attributes. These values can change while the program is running
-- Objects combined data with the operations on that data (Back 2013, p. 11)
-- Objects as processes (Back 2013, p. 12)

2) Behaves appropriately when called upon

3) Objects as abstraction (abstract view)

https://cdn.rawgit.com/AUAP/AP2017/18dfb50f/class08/sketch08/index.html

Advantages of O-O programming

- Make code easier to understand and reuse in other contexts
  - Allows the programmer to refine the code for each object independently from the entire program
  - Inheritance

Implementation

```
// step 1. Declare Objects
var car = [];


//step 2. Initialize object
 car[0] = new Car(color(255,0,0), 10, 10, 100, 20);
 car[1] = new Car(color(0,0,255), 20, 20, 300, 10);


//step 3. Use object
 for (var i = 0; i <car.length; i++) {
  car[i].drive();
  car[i].display();
 }
```

```
function Car(getcolor, speed, xpos, ypos, size) {
        this.getcolor = getcolor;
        this.speed = speed;
        this.xpos = xpos;
        this.ypos = ypos;
        this.size = size;

        this.display = function() {
                stroke(0);
                fill(this.getcolor);
                rect(this.xpos,this.ypos,this.size,this.size);
        }

        this.drive = function() {
                this.xpos = this.xpos + this.speed;
                if (this.xpos > width) {
                        this.xpos = 0;
                }

        }

}
```

https://github.com/AUAP/AP2017/blob/master/class08/sketch08/sketch.js

Mini_exe7- Create your own Form Art