Parametric Design and Visual Programming

Moran Mizrahi
moran.mizrahi@mail.huji.ac.il
Outline

- Motivation
- Parametric Design Definition
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Outline

- **Motivation**
- Parametric Design Definition
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Motivation

- Designers often use computer-aided tools to build models and help them visualize ideas. However, the vast majority of these models are still built in such a way that they are difficult to modify interactively.

- The problem becomes more severe when bespoke 3D models are geometrically complex. Changing one aspect of such a model usually requires extensive low-level modifications to many of its other parts.

- We would like to allow the designer to specify relationships among various parameters of their design model, and then let them change these parameters in a way that the remainder of the model will react and update accordingly.

- This kind of design both requires and helps to create powerful interactive tools that allow designers to explore and optimize a multitude of possibilities while reducing the amount of time it takes to do so in a rigorous manner.
Outline

- Motivation
- **Parametric Design Definition**
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Parametric Design definition

“Parametric design is a process based on algorithmic thinking that enables the expression of parameters and rules that, together, define, encode and clarify the relationship between design intent and design response”

(Parametric design for architecture, Wassim Jaby)
Outline

- Motivation
- Parametric Design Definition
- General examples
  - Fundamental themes in Parametric Design
  - Parametric Patterns
  - Parametric Design tools
Aviva Stadium, Dublin, Ireland.

Aviva Stadium was designed to replace an earlier stadium located on the same site and to provide a modern facility that could accommodate 50,000 seated spectators and the home games of the Irish international Rugby and the Irish Football teams. A project requirement, from the outset, was to respect the historic site of Lansdowne Road by minimizing the impact of the new facility on surrounding buildings.
Aviva Stadium, Dublin, Ireland (2)
Kinematics – Nervous System
Outline

- Motivation
- Parametric Design Definition
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Fundamental themes in Parametric Design

(1) Versioning
(2) Iteration
(3) Mass customization
(4) Continuous differentiation

Parametricism - A new global style for architecture and urban design, Shumacher Patrick.
Fundamental themes in Parametric Design

(1) Versioning

The term refers to the process of creating versions – or variations on a certain design solution based on varying conditions.

Parametric software allows the designer to create a prototype solution that, rather than being cast in a static CAD file format, is wired. This wiring allows the design solution to be tweaked and manipulated, creating new versions when new forces and conditions arise.
Fundamental themes in Parametric Design

(2) Iteration
The term refers to cycling through or repeating a set of steps. Iteration can create variation at every pass through the same set of instructions.
Fundamental themes in Parametric Design

(3) Mass customization

One of the main successes of the industrial revolution is the idea of mass production. Given the advent of digital fabrication technologies, we are now able to change the manufacturing instructions between each object. Given that the process is parameterized and robotic, it often costs the same to mass-customize the manufactured products as it does to mass-produce the same quantity of identical products.
Fundamental themes in Parametric Design

(4) Continuous differentiation
Parametrically varied instances within an overall group, curve or field maintain their continuity to other instances before and after them while uniquely responding to local conditions.
Outline

- Motivation
- Parametric Design Definition
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Parametric Patterns

(1) Controller
(2) Force field
(3) Repetition
(4) Tiling
(5) Recursion
(6) Subdivision
(7) Packing
(8) Weaving
(9) Branching
Parametric Systems – (1) Controller

Parametric systems allow us to create rigs that control the overall design by modifying a few of their controlling parameters. The basic concept behind the controller pattern is that it separates and clarifies the process by which the main model will change.
A prevalent pattern in parametric design algorithms is that of the force field. When thinking about what affects the form of an object or building, it is natural to imagine the metaphor of various forces that are pushing and pulling on it. These vectors rhythmically change direction, shape or intensity based on their location in the force field and the presence or absence of various forces acting on them.
Repetition can be thought of as the simple act of copying an element multiple times. In parametric systems, repetition can become more interesting, because a repeated element can maintain the basic topology of its predecessor without having to be exactly identical to it.
Parametric Patterns – (4) Tiling

In mathematics, tiling is defined as the arrangement of identical planar shapes to completely cover a given area without overlapping. One can think of tiling as a natural extension of the concept of repetition, but in 2D. Parametrically, any aspects of the tile can be varied as it repeats.
Recursion is a special case of repetition in which the repetition is achieved by having a process call upon itself to generate the next iteration. Formally, recursion is defined as the process of repeating items in self-similar ways.

Genetic stairs
In many cases, designers working with smooth surfaces and forms need to subdivide them in order to unfold them into planar components that can be digitally fabricated on CNC machines or laser cutters. Subdivision is the process of separating a continuous surface into smaller components by tracing, scoring or cutting lines through the surface.
Closely related to the concepts of tiling and subdivision is the concept of packing: the placement of many objects in a space, such that little or nothing of it is left over (in nature: pomegranate). Components in a packing systems seek empty space to occupy.
Weaving was known in the Palaeolithic era and continues to this day. In its simplest form, weaving creates a fabric from the interlacing of two threads at right angles to each other. This method gives a surface structural strength.
Branching is a basic topological growth mechanism in nature for maximizing surface area, chanelling resources and responding to structural forces. Branching systems in nature are self-similar and recursive and so tend to be fractal.
Outline

- Motivation
- Parametric Design Definition
- General examples
- Fundamental themes in Parametric Design
- Parametric Patterns
- Parametric Design tools
Parametric Design tools

(1) Para
(2) Processing
(3) Grasshopper
Para is a digital illustration tool that uses direct manipulation to define and edit procedural artwork. Through creating and altering vector paths, artists can define iterative distributions and parametric constraints.

Para began as opensource software in the Creative Technologies Lab at Adobe Research. Currently, Para is under development in the Lifelong Kindergarten Group at the MIT Media Lab.
Parametric Design tools – (2) Processing

- processing is a free and open-source Java-based programming environment for visual designers.
- Its ease of use and simple interface make it suitable for the beginner scripter. It excels in 2D visual design without the need for any external plug-ins.

Download Processing
Processing tutorials
Connecting Arduino to Processing
Parametric Design tools – (2) Processing

Necessary function:

```java
public void setup() {
}

public void draw() {
}
```

Optional functions:

```java
public void keyPressed() {
}

public void mousePressed() {
}

public void mouseDragged() {
}

public void mouseReleased() {
}
```
Parametric Design tools – (2) Processing

```cpp
// Example: Two Car objects
Car myCar1;
Car myCar2; // Two objects!

void setup() {
  size(200,200);
  // Parameters go inside the parentheses when the object is constructed.
  myCar1 = new Car(color(255,0,0), 0, 200, 2);
  myCar2 = new Car(color(0,0,255), 0,10,1);
}

void draw() {
  background(255);
  myCar1.drive();
  myCar1.display();
  myCar2.drive();
  myCar2.display();
}

// Even though there are multiple objects, we still only need one class.
// No matter how many cookies we make, only one cookie cutter is needed.

class Car {
  color 0;
  float xpos;
  float ypos;
  float xspeed;

  // The Constructor is defined with arguments.
  Car(color tempC, float tempXpos, float tempYpos, float tempXspeed) {
    c = tempC;
    xpos = tempXpos;
    ypos = tempYpos;
    xspeed = tempXspeed;
  }

  void display() {
    stroke(0);
    fill(0);
    rectMode(CENTER);
    rect(xpos, ypos, 20, 10);
  }

  void drive() {
    xpos += xspeed;
    if (xpos > width) {
      xpos = 0;
    }
  }
}
```
Parametric Design tools – (3) Grasshopper

Grasshopper is a visual programming language developed by David Rutten at Robert McNeel & Associates. Grasshopper runs within the Rhinoceros 3D CAD application. Programs are created by dragging components onto a canvas. The outputs to these components are then connected to the inputs of subsequent components. Grasshopper is primarily used to build generative algorithms.

Download Grasshopper
Tutorial videos
Free online books
Parametric Design tools – (3) Grasshoppe
Parametric Design tools – (3) Grasshopper

Connecting Arduino to Grasshopper (should download Firefly plugin for Grasshopper first):

Download Firefly
Connecting Arduino to Grasshopper
## Parametric Design tools comparison

<table>
<thead>
<tr>
<th></th>
<th>Para</th>
<th>Grasshopper</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animation</td>
<td>נייד</td>
<td>לא</td>
<td>כן</td>
</tr>
<tr>
<td>Requires 2D</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Geometric</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Requires high</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Requires basic</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Includes code</td>
<td>כן</td>
<td>כן</td>
<td>לא</td>
</tr>
<tr>
<td>Requires plan</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>High fidelity</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Easy to use</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>User friendly</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Accessible</td>
<td>כן</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>para to controllers</td>
<td>כן</td>
<td>כן</td>
<td>לא</td>
</tr>
<tr>
<td>Suitable for</td>
<td>כן</td>
<td>כן</td>
<td>כן</td>
</tr>
<tr>
<td>Parameters</td>
<td>לא</td>
<td>לא</td>
<td>לא</td>
</tr>
<tr>
<td>Options many</td>
<td>כן</td>
<td>כן</td>
<td>לא</td>
</tr>
<tr>
<td>Change parameters easily</td>
<td>כן</td>
<td>כן</td>
<td>לא</td>
</tr>
</tbody>
</table>