

Processing

um breve panorama

Educação em Arte e Tecnologia

Luiz Ernesto Merkle
merkle@utfpr.edu.br

Slides baseados na palestra apresentada no FISL 9.0, Porto Alegre
Programa de Pós-Graduação em Tecnologia
Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná
e
Comunidade Yguá de Pesquisa em Software Livre



Processing

um breve panorama

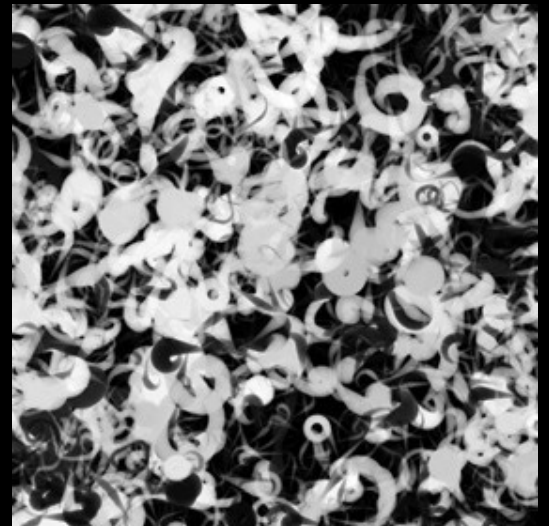
Luiz Ernesto Merkle
merkle@utfpr.edu.br

Esta palestra tem por objetivo apresentar as principais características do ambiente e da linguagem Processing, assim como ilustrar por meio de alguns exemplos suas potencialidades.

Processing tem sido usada por designers, artistas, hobistas e demais interessadas/dos em aplicações que vão da internet, passam por interagir com plataformas de hardware aberto, e chegam a instalações de arte, as vezes simultaneamente, dentre outras.

Para saber mais sobre a linguagem/ambiente ver
<http://www.processing.org>

A high-contrast, black and white photograph of a dense, textured surface. The image shows intricate, overlapping, and somewhat crystalline structures, possibly a close-up of a plant or a microscopic view. The lighting creates deep shadows and bright highlights, emphasizing the complex geometry and organic forms of the material.



Ben Fry (<http://benfry.com/>)

Ben Fry received his doctoral degree from the Aesthetics + Computation Group at the MIT Media Laboratory, where his research focused on combining fields such as Computer Science, Statistics, Graphic Design, and Data Visualization as a means for understanding complex data. [...] During the 2006-2007 school year, Ben was the Nierenberg Chair of Design for the the Carnegie Mellon School of Design. He currently works as a designer in Cambridge, MA. His personal work has shown at the Whitney Biennial in 2002 and the Cooper Hewitt Design Triennial in 2003. Other pieces have appeared in the Museum of Modern Art in New York, at Ars Electronica in Linz, Austria and in the films "Minority Report" and "The Hulk." His information graphics have also illustrated articles for the journal Nature, New York Magazine, and Seed.

ben fry - Iceweasel

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

← → ↺ ↻ ↵ B http://benfry.com/ Google

Getting Started Latest Headlines Utfpr ppgte DAINF Design e Semiotica ... Oficinas de Integração

Ben Fry

/ projects / writing / data / genetics / about

Books

Visualizing Data
I recently finished *Visualizing Data* for O'Reilly. It was published in December 2007 and can be found at Amazon and elsewhere. The book covers the data visualization process (Chapter 1), and a couple projects you'll find on this site, such as Salary vs. Performance (Chapter 4), Zipdecode (Chapter 6), and Anemone (Chapter 8). More about the book can be found in the [writing](#) section of the site.

Processing
The Processing book from MIT Press is now available! Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production.

News

4 March 08
Finally, a new web site!

24 February 08
Redesigned diagram of the genetic code, including an interactive version

5 May 07
Salary vs. Performance sketch updated for 2007 season

20 April 07
Celebrating the one year anniversary since an update.. sorry, more soon

19 April 06
salary vs. performance sketch updated for 2006 season

10 April 06
Quick illustration for New York Magazine's high priority feature

20 February 06
Two illustrations for a New York Magazine cover story about blogs

31 October 05
Isometric clocks depicting the HapMap on the cover of Nature

31 October 05
A poster (PDF) about the HapMap project for Genome

Recent

24 February–12 May 08
Pieces in Design and the Elastic Mind at the MoMA (and several Processing-based projects from others too!)

22 February 07
Pecha Kucha Boston 03

21 February 07
Visit a class and lecture at Smith College

28 September 07–6 January 08
Processing travels with the National Design Triennial to the ICA Boston

13 December 07
ICA/AIGA Design Series: Chip Kidd, Ben Fry, Toshiko Mori hosted by Ellen Lupton (canceled due to weather)

14–28 November 07
Visualizar workshop in Madrid

28 September 07
Processing talk for PICNIC Festival in Amsterdam

21 September 07
Simply Science Simply Design symposium in Sunderland

6 September 07
Kevnote at Ignite Boston, an

Upcoming

11 April 08
Lecture at UNC Asheville

19 April 08
SEE Conference in Wiesbaden, Germany

28 April 08
Visualization lecture for a Harvard course

2 May 08
Speaking at Williams College

16 May 08
Processing lecture at MIT CSAIL

20 May 08
Lecture at Virginia (UVA)

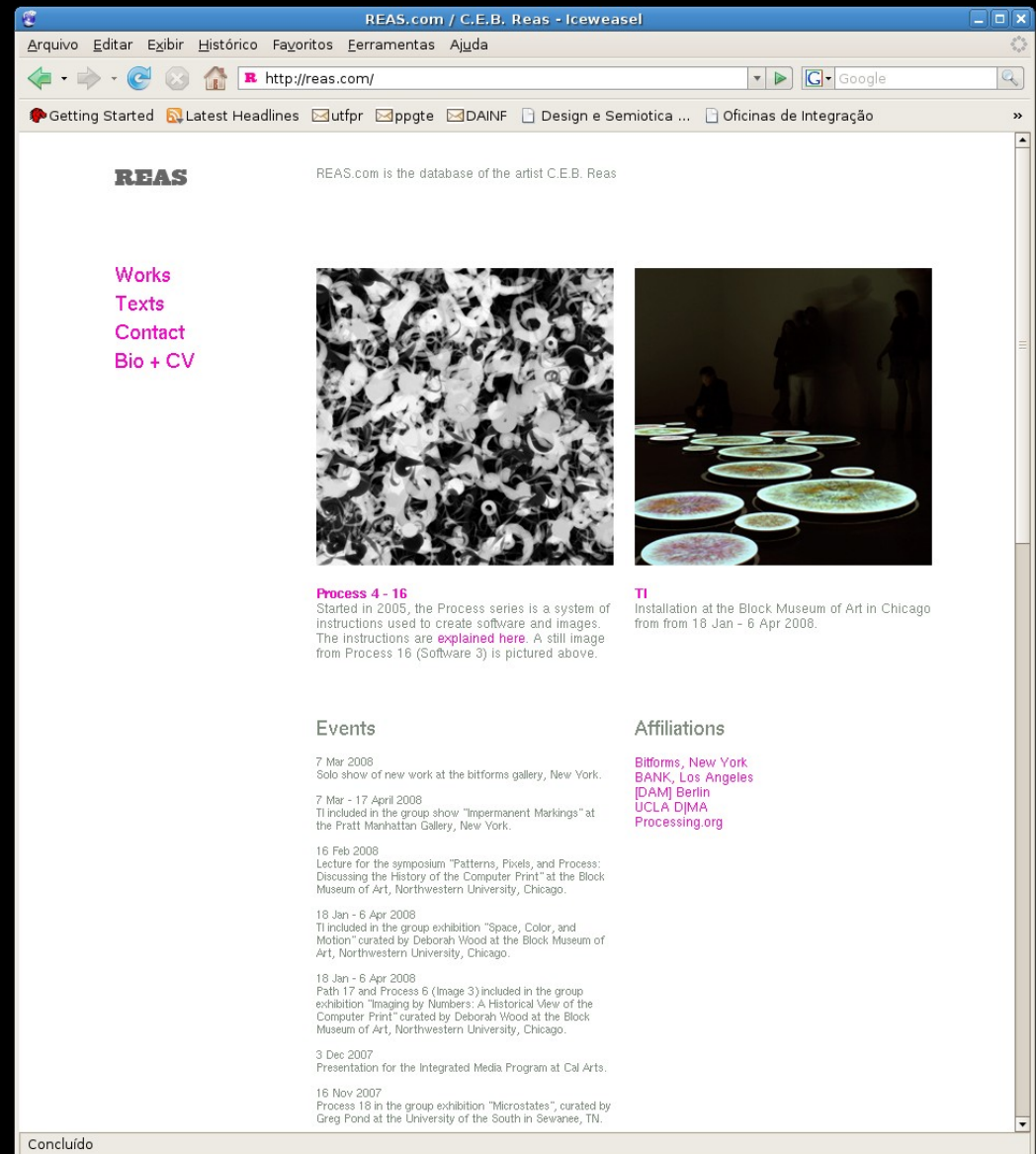
21 August 08
iCORE Summit 08 in Banff

afisl2008.odp - BrOffice.org Impress ben fry - Iceweasel [Capturando Imagens da Tela - Iceweasel]

Casey Reas (<http://reas.com/>)

C.E.B. Reas is an artist who lives and works in Los Angeles. He focuses on defining processes and translating them into images. He is an associate professor and chair of the department of Design |Media Arts at the University of California, Los Angeles.

He has exhibited his work internationally at institutions including Laboral (Gijon, Spain), The Cooper-Hewitt Museum (New York), and the National Museum for Art, Architecture, and Design (Oslo), at independent venues including Telic Arts Exchange (Los Angeles), <>TAG (The Hague), and Ego Park (Oakland), at galleries including Bitforms (New York), BANK (Los Angeles), and [DAM] Berlin, and at festivals including Sonar (Barcelona), Ars Electronica (Linz), and Microwave (Hong Kong).

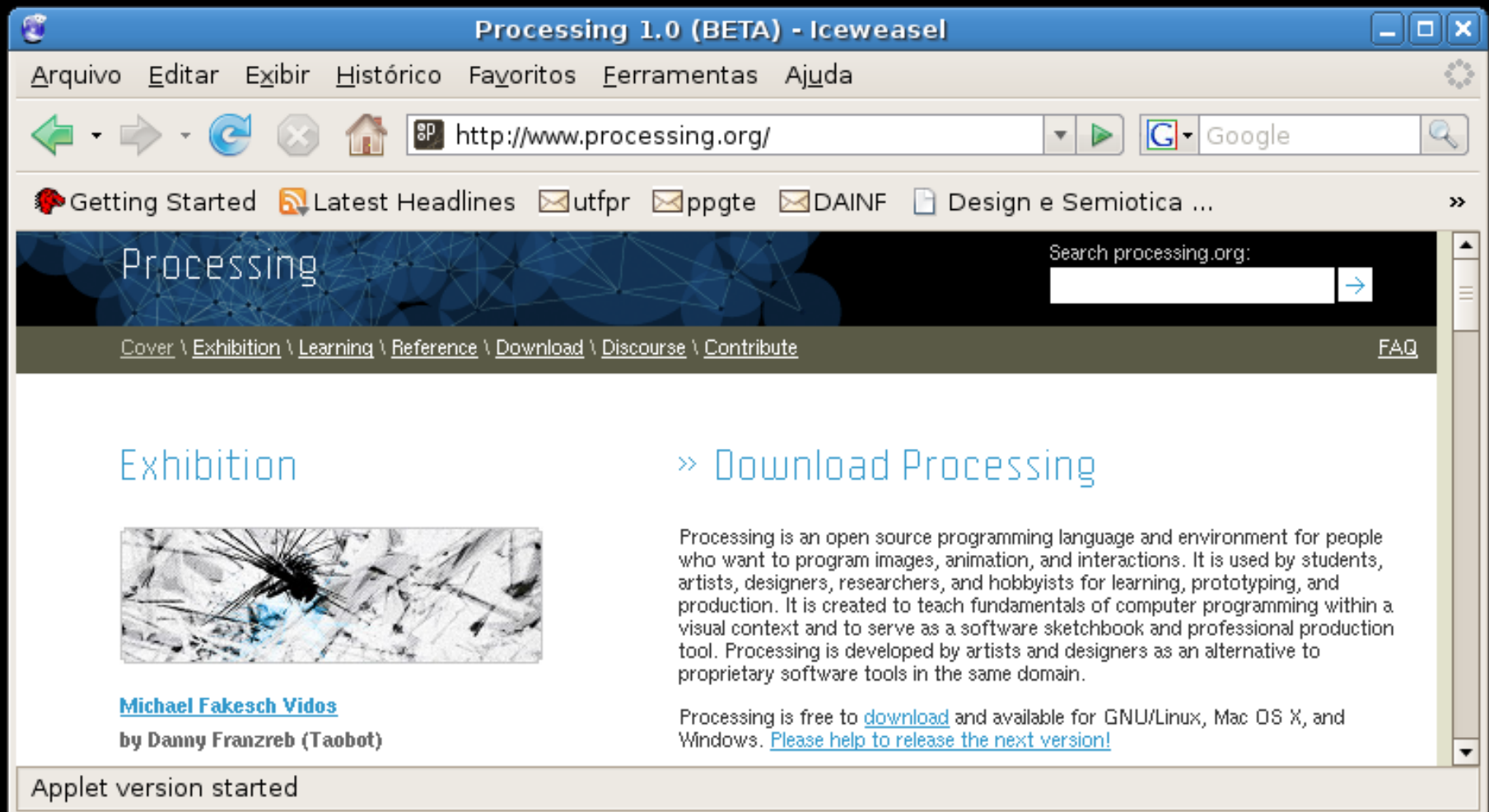


Fry, em seu website antigo se refere a processing como:

1. um **sítio web** [<http://www.processing.org/>]
2. um ambiente de programação para se aprender design computacional;
3. um bloco de rascunho para se prototipar rapidamente trabalhos do tipo incluídos aqui [<http://acg.media.mit.edu/people/fry/>];
4. uma nova API gráfica e uma nova máquina de renderização para Java;
5. um projeto aberto iniciado por casey reas e eu mesmo [ben fry];
6. uma comunidade ativa de umas poucas milhares de pessoas que estão utilizando o software.

Fry, em seu website antigo se refere a processing como:

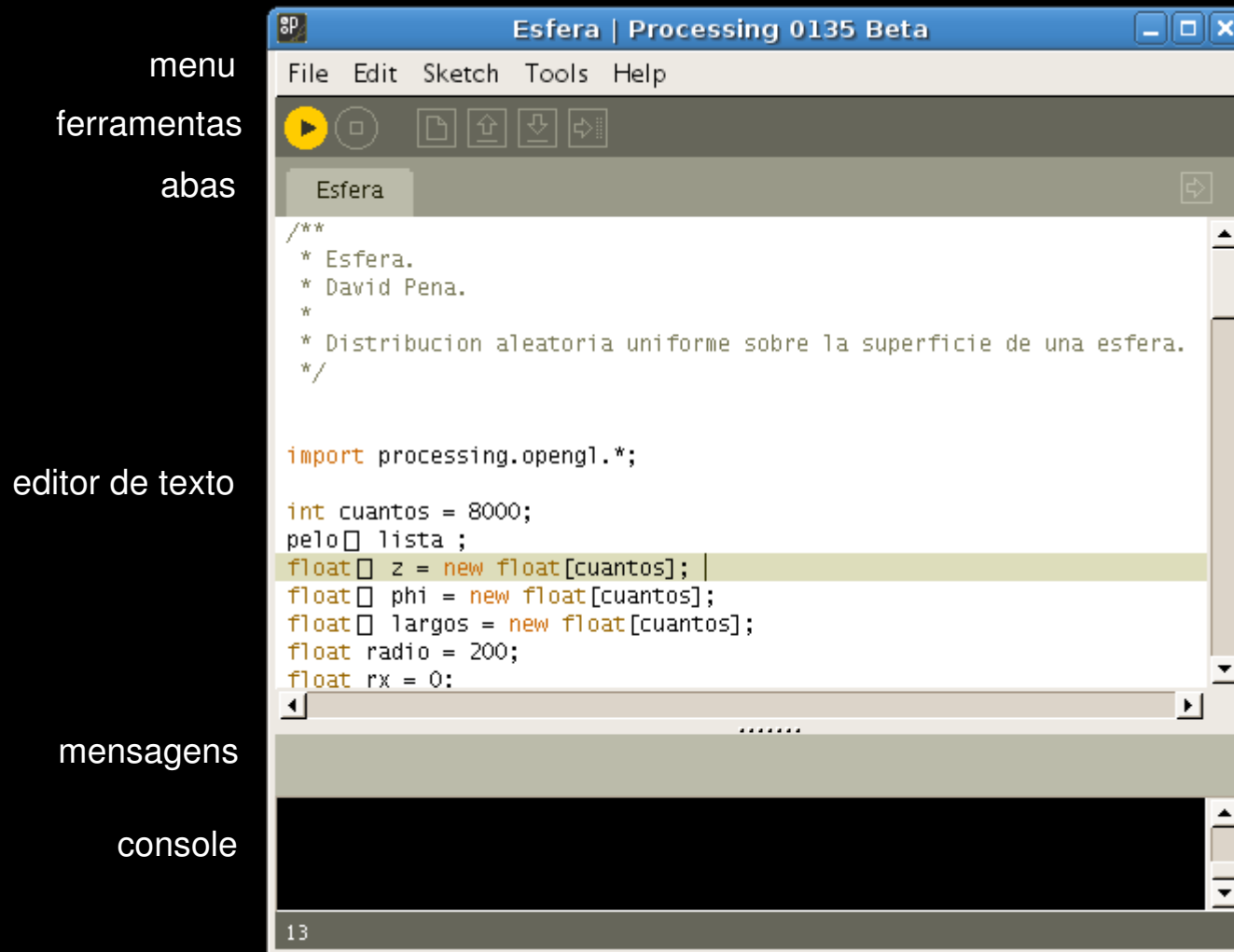
1. um **sítio web** [http://www.processing.org/];



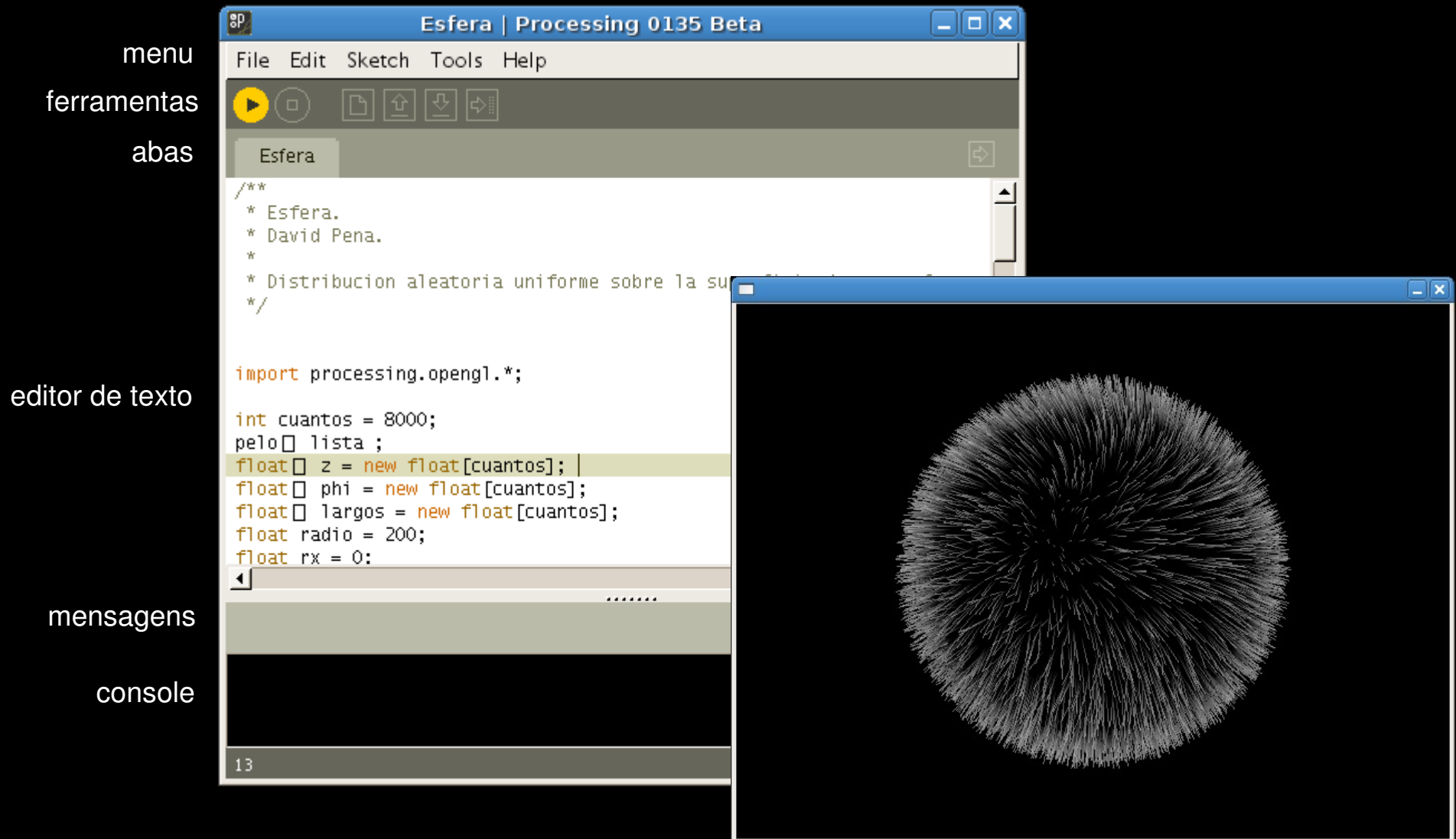
Fry, em seu website antigo se refere a processing como:

1. um sítio web [<http://www.processing.org>]
2. um **ambiente de programação** para se aprender design computacional;
3. um **bloco de rascunho** para se prototipar rapidamente trabalhos do tipo incluídos aqui [<http://acg.media.mit.edu/people/fry/>];
4. uma nova API gráfica e uma nova máquina de renderização para Java;
5. um projeto aberto iniciado por casey reas e eu mesmo [ben fry];
6. uma comunidade ativa de umas poucas milhares de pessoas que estão utilizando o software.

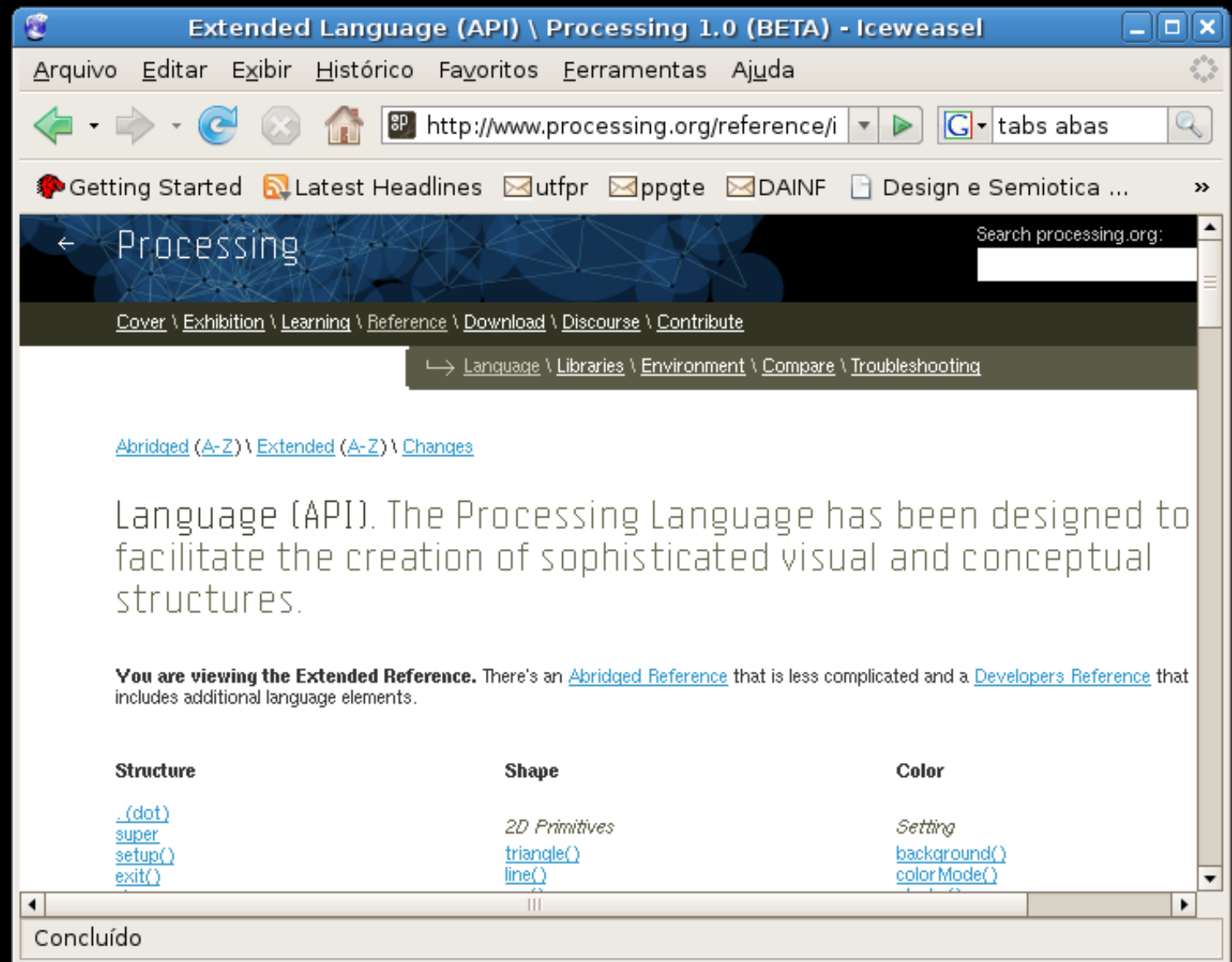
2. um ambiente de programação
3. um bloco de rascunho



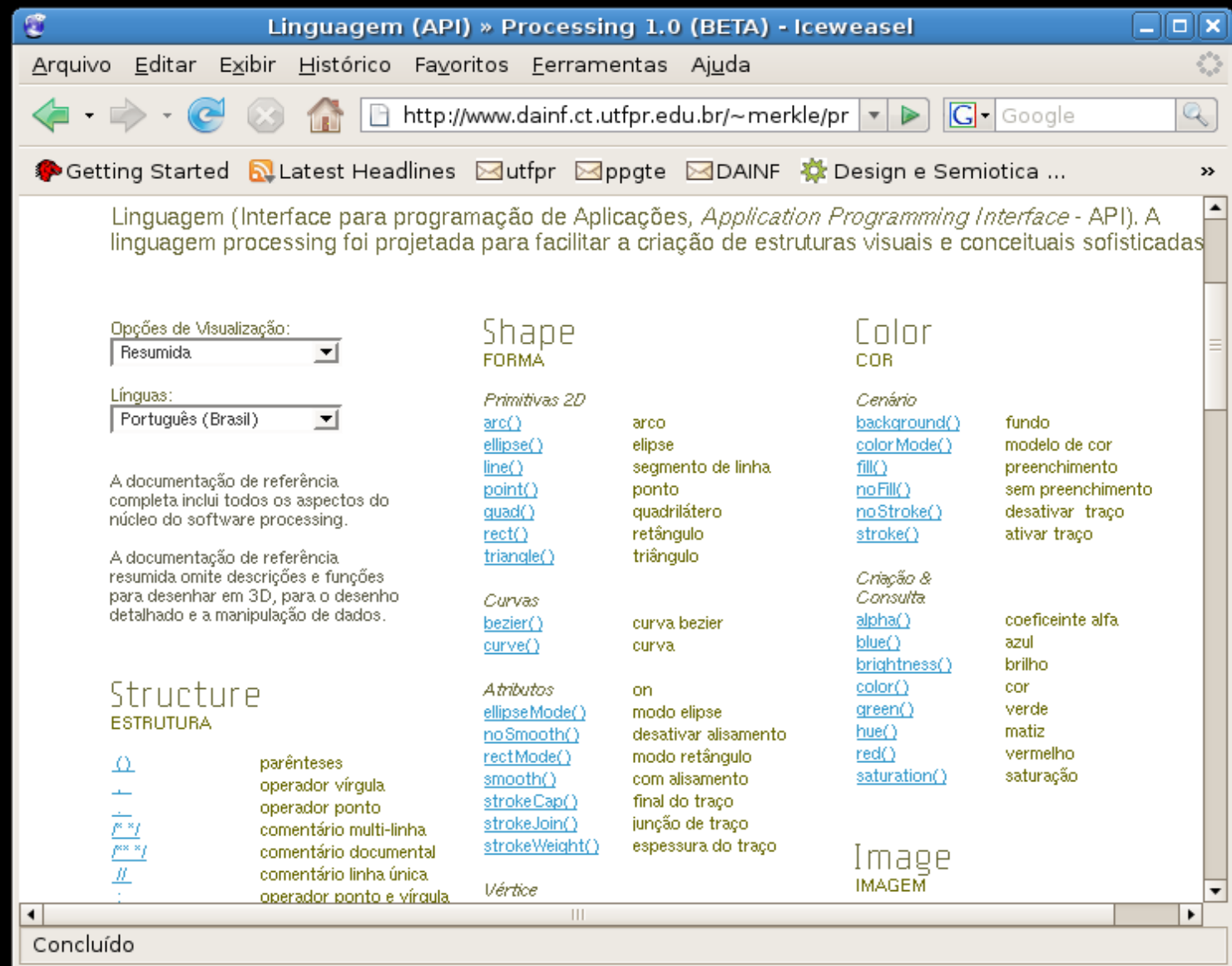
2. um ambiente de programação
3. um bloco de rascunho



4. uma nova **API gráfica** e uma nova **máquina de renderização** para Java;



4. uma nova **API gráfica** e uma nova **máquina de renderização** para Java;



tradução da versão 91+:
de novembro de 2005

<http://www.dainf.ct.utfpr.edu.br/~merkle/processing/reference/ptBR/index.html>

4. uma nova **API gráfica** e uma nova **máquina de renderização** para Java;

Structure

. (dot)
super
setup()
exit()
class
void
size()
/* */ (multiline comment)
false
noLoop()
this
true
new
// (comment)
, (comma)
delay()
{ } (curly braces)
/** */ (doc comment)
draw()
loop()
implements
redraw()
; (semicolon)
[] (array access)
extends
null
() (parentheses)
return
= (assign)

Environment

frameRate()
online
focused
noCursor()
frameRate
screen
width
height
frameCount
cursor()

Data

Primitive

color
char
float
int
boolean
byte

Composite

String
Array
Object

Conversion

unbinary()
char()
int()
hex()
binary()
unhex()
byte()
str()
boolean()
float()

String Functions

split()
join()
splitTokens()
nf()
match()
trim()
nfc()
nfs()
nfp()

Array Functions

shorten()
concat()
subset()
append()
sort()
arraycopy()
reverse()
splice()
expand()

Control

Relational Operators

< (less than)
>= (greater than or equal to)
== (equality)
> (greater than)
<= (less than or equal to)
!= (inequality)

Iteration

for
while

Conditionals

?: (conditional)
break
default
switch()
case
else
if

Logical Operators

|| (logical OR)
&& (logical AND)
! (logical NOT)

4. uma nova **API gráfica** e uma nova **máquina de renderização** para Java;

Shape

2D Primitives

triangle()
line()
arc()
point()
quad()
ellipse()
rect()

Curves

bezierDetail()
bezierTangent()
curveTightness()
bezierPoint()
curveDetail()
curvePoint()
curve()
bezier()

3D Primitives

box()
sphere()
sphereDetail()
Attributes
strokeWeight()
smooth()
strokeJoin()
noSmooth()
ellipseMode()
rectMode()
strokeCap()

Vertex

vertex()
bezierVertex()
textureMode()
beginShape()
texture()
curveVertex()
endShape()

Input

Mouse

mouseDragged()
mouseMoved()
mouseButton
mouseX
mouseReleased()
pmouseX
mousePressed()
mouseY
mousePressed
pmouseY
mouseClicked()

Keyboard

keyTyped()
keyCode
keyReleased()
keyPressed()
key
keyPressed

Files

openStream()
open()
loadStrings()
loadBytes()
Web
status()
link()
param()

Time & Date

hour()
millis()
year()
minute()
month()
day()
second()

Output

Text Area

println()
print()

Image

saveFrame()
save()

Files

endRecord()
saveStrings()
saveBytes()
createWriter()
PrintWriter
createReader()
beginRecord()

Transform

rotateY()
printMatrix()
pushMatrix()
rotateZ()
applyMatrix()
scale()
popMatrix()
translate()
resetMatrix()
rotate()
rotateX()

Lights, Camera

Lights

noLights()
directionalLight()
lightFalloff()
pointLight()
lights()
lightSpecular()
ambientLight()
normal()
spotLight()

Camera

camera()
ortho()
endCamera()
printCamera()
perspective()
frustum()
printProjection()
beginCamera()

Coordinates

modelZ()
screenZ()
modelX()
screenX()
modelY()
screenY()

Material

Properties

shininess()
specular()
ambient()
emissive()

4. uma nova **API gráfica** e uma nova **máquina de renderização** para Java;

Color

Setting

background()
colorMode()
stroke()
noFill()
noStroke()
fill()

Creating & Reading

blendColor()
red()
brightness()
blue()
saturation()
lerpColor()
green()
hue()
alpha()
color()

Image

PImage
createImage()

Loading & Displaying

loadImage()
image()
noTint()
imageMode()
tint()

Pixels

filter()
copy()
set()
updatePixels()
blend()
loadPixels()
pixels[]
get()

Rendering

PGraphics
hint()
unhint()
createGraphics()

Typography

PFont
*Loading
& Displaying*

text()
createFont()
loadFont()
textFont()

Attributes

textMode()
textSize()
textAlign()
textLeading()
textWidth()

Metrics

textDescent()
textAscent()

Math

Operators

-= (subtract assign)
+ (addition)
- (minus)
% (modulo)
+= (add assign)
/ (divide)
* (multiply)
++ (increment)
-- (decrement)

Bitwise

Operators

& (bitwise AND)
<< (left shift)
| (bitwise OR)
>> (right shift)

Calculation

min()
max()
round()
dist()
exp()
pow()
floor()
sqrt()
abs()
constrain()
norm()
mag()
log()
lerp()
sq()
ceil()
map()

Trigonometry

acos()
tan()
sin()
cos()
degrees()
atan2()
atan()
radians()
asin()

Random

noise()
noiseSeed()
randomSeed()
noiseDetail()
random()

Constants

HALF_PI (1.57079...)
TWO_PI (6.28318...)
PI (3.14159...)

5. um projeto aberto (GPL e LGPL);

Processing is Open Source Software.

The PDE (Processing Development Environment) is released under the GNU GPL (General Public License).

The export libraries (also known as 'core') are released under the GNU LGPL (Lesser General Public License).

There's more information about Processing and Open Source in the FAQ and more information about the GNU GPL and GNU LGPL at opensource.org.

Please contribute to Processing! (<http://processing.org/download/index.html>)

Processing é um software de código aberto.

O PDE (Ambiente de Desenvolvimento do Processing) liberado como Licença Pública Geral (GNU GPL).

As bibliotecas de exportação (também conhecidas como do 'núcleo') são liberadas sob Licença Pública Geral Menor (GNU LGPL).

Há mais informações sobre o Processing e Código Aberto em seu FAQ e mais informações sobre a GNU GPL e a GNU LGPL em opensource.org.

Favor contribuir com o Processing! (tradução própria, grifo meu).

6. uma comunidade (25853 membros inscritos em 11/04/08);

The screenshot shows a web browser window titled "Processing 1.0 (BETA) - Index - Iceweasel". The address bar shows the URL "http://processing.org/discourse/yabb_beta/YaBB.cgi". The browser's menu bar includes "Arquivo", "Editar", "Exibir", "Histórico", "Favoritos", "Ferramentas", and "Ajuda". The page header features a navigation bar with links: "Getting Started", "Latest Headlines", "utfpr", "ppgte", "DAINF", "Design e Semiotica ...", and "Oficinas de Integração". Below this is a banner for "Processing" with a search bar labeled "Search processing.org:". A breadcrumb trail reads: "Cover \ Exhibition \ Learning \ Reference \ Download \ Discourse \ Contribute".

Below the banner, there are links: "Index \ Search \ Members \ Profile \ Notification \ Logout \ RSS". A text block states: "The board for Processing (BETA) launched 20 April 2005. You can still visit the [the \(ALPHA\) board](#). Hey, Luiz Merkle, you have [0 messages](#)."

The main content area is titled "Processing 1.0 (BETA)" and contains a table with forum categories. The table has columns: "Forum name", "Topics", "Posts", and "Last post".

Forum name	Topics	Posts	Last post
Discussion			
Community, Collaboration, Status Interface with the Processing developers and read about the current status Moderators: fry , REAS	273	1203	Today at 6:23am by jungalero
Events, Publications, Opportunities Processing related conferences, courses, workshops, concerts, articles, jobs, etc. Moderators: fry , REAS	209	567	Apr 9 th , 2008, 10:11pm by REAS
Exhibition What are you making with Processing? This is a place to share your finished work and work-in-progress for open discussion. Moderators: fry , REAS	298	1026	Apr 10 th , 2008, 10:46am by ruipenha
Programming Questions & Help			
Syntax			

At the bottom left of the browser window, the text "Concluído" is visible.

6. uma **comunidade** (25853 membros inscritos em 11/04/08);

Discussão:

- Comunidade, Colaboração, Status
- Eventos, Publicações, Oportunidades
- Exibição

Ajuda e Questões de Programação

- Sintaxe
- Programas
- OpenGL
- Som, Música
- Capturas de Vídeo, Filmes
- Eletônica
- Outras Bibliotecas
- Integração

Sugestões e Bugs:

- Software, Sugestões ao Sítio Web
- Bugs de Software
- Website, Documentação, Books Bugs

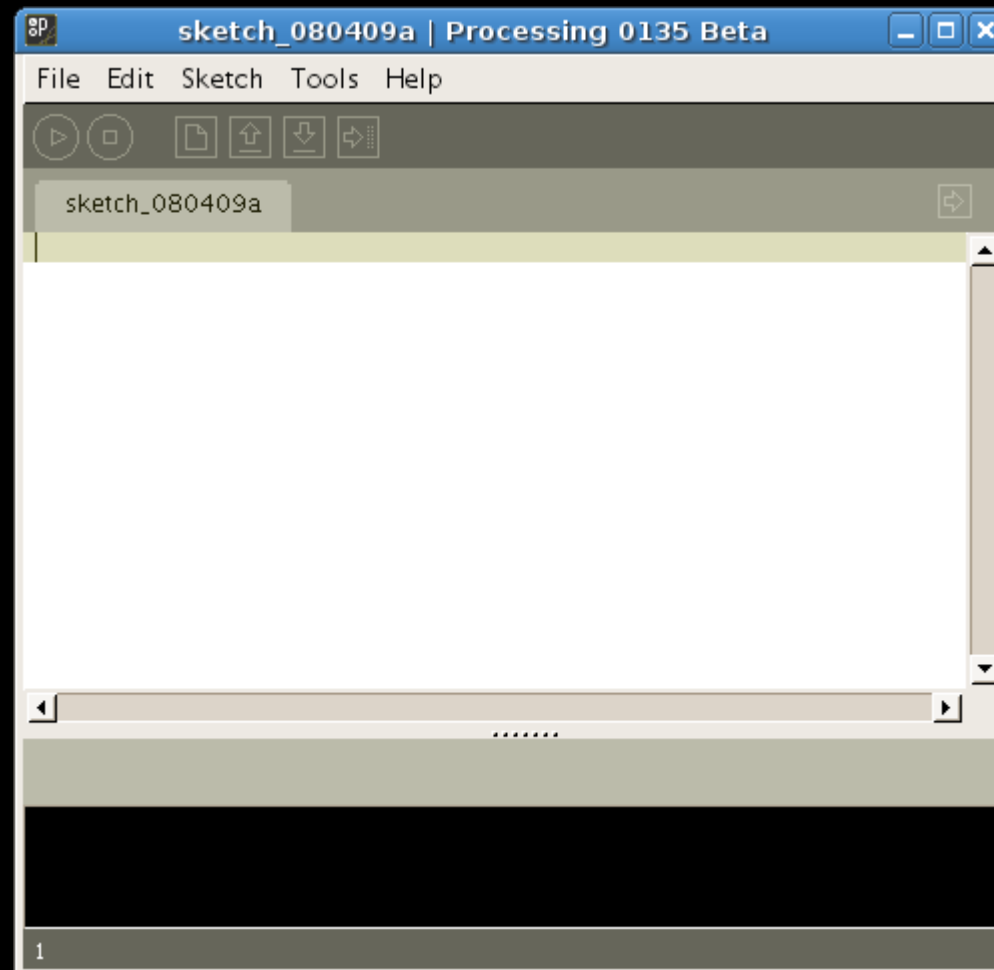
Código Aberto, Processing Dev

- Bibliotecas, Desenvolvimento de Ferramentas
- Núcleo, Ambiente de Desenvolvimento (PDE)

<http://processing.org/discourse/index.html>

Utilizando Processing

Arquivo	Editar	Rascunhar	Ferramentas	Ajuda	
Executar	Parar	Novo	Abrir	Salvar	Exportar



Utilizando Processing

Um primeiro modo (comumente **linear** e **introdutório**)

Utilizando Processing

Um primeiro modo (mais simples)



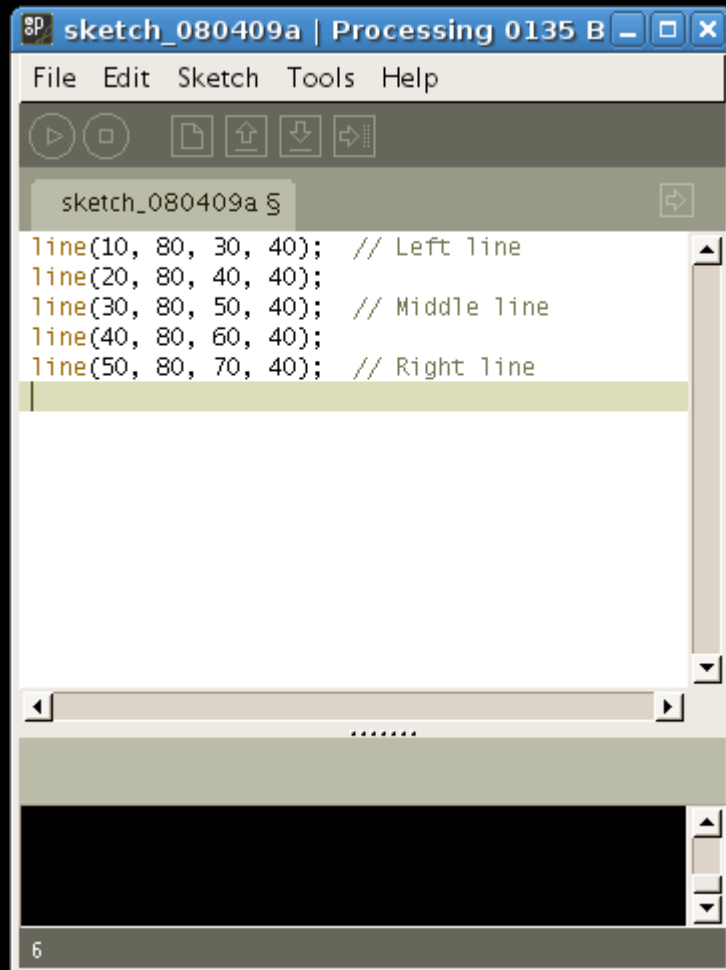
// Formas geométricas

```
line(10, 80, 30, 40); // Left line  
line(20, 80, 40, 40);  
line(30, 80, 50, 40); // Middle line  
line(40, 80, 60, 40);  
line(50, 80, 70, 40); // Right line  
// Reas e Fry( 2007, p.11, 0-01)
```

```
// Abra o ambiente processing (PDE)  
// Abra um novo esboço  
// copie o exemplo  
// rode o esboço
```

Utilizando Processing

Um primeiro modo (mais simples)



// Formas geométricas

`line(10, 80, 30, 40);` // Left line

`line(20, 80, 40, 40);`

`line(30, 80, 50, 40);` // Middle line

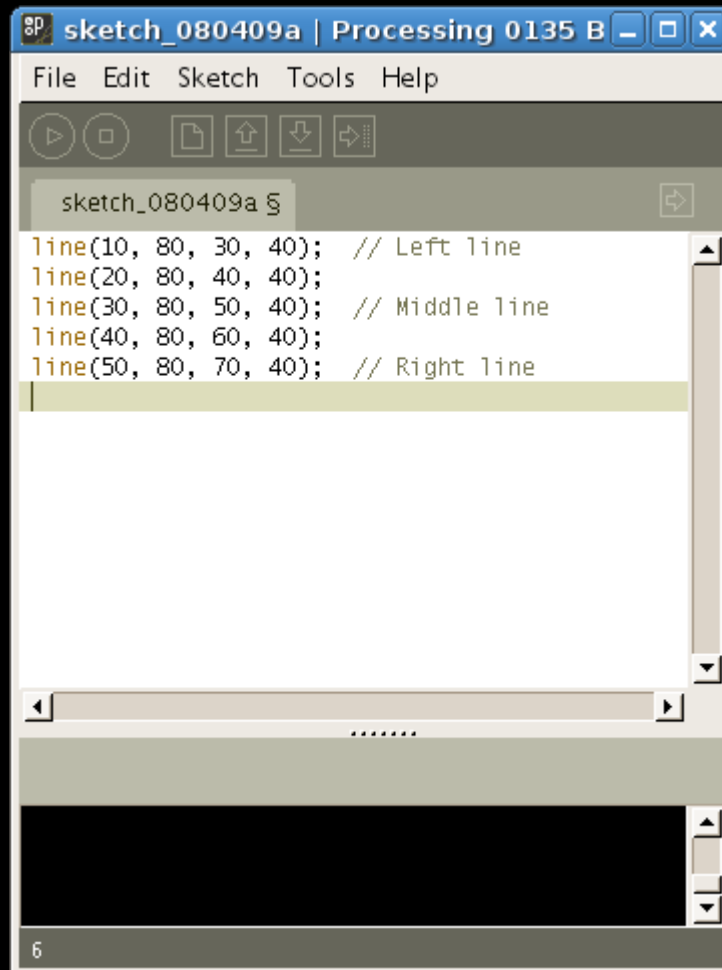
`line(40, 80, 60, 40);`

`line(50, 80, 70, 40);` // Right line

// Reas e Fry(2007, p.11, 0-01)

Utilizando Processing

Um primeiro modo (mais simples)



// Formas geométricas

```
line(10, 80, 30, 40); // Left line
line(20, 80, 40, 40);
line(30, 80, 50, 40); // Middle line
line(40, 80, 60, 40);
line(50, 80, 70, 40); // Right line
// Reas e Fry( 2007, p.11, 0-01)
```


Utilizando Processing

Um primeiro modo (**mais simples**)

O modo **mais simples** de programar em processing permite à/ao estudante compreender conceitos simples de computação, necessários ao desenvolvimento de mídias digitais como:

- o **armazenamento de dados** em variáveis numéricas e textuais;
- a **atribuição de valores** à variáveis ao longo da execução de um esboço;
- a **temporalidade envolvida na execução de uma instrução** após a outra, o que pode ser feito envolvendo o desenho de várias formas sobrepostas;
- o **controle do fluxo de execução** de um esboço por meio de condicionais (if else) e laços (for, while);
- o uso de **bibliotecas** em diversos domínios, inclusive **gráficas**;
- ter contato com **exemplos mais complexos**;

Com muitos exemplos que exploram recursos visuais significativos, a/o estudante logo se ve motivada/o para desenvolver esboços que dariam menos trabalho se programados com recursos apropriados. Vejamos alguns exemplos.

Utilizando Processing

```
// cores
background(0);
noStroke();
smooth();
fill(242, 204, 47, 160); // Yellow
ellipse(47, 36, 64, 64);
fill(174, 221, 60, 160); // Green
ellipse(90, 47, 64, 64);
fill(116, 193, 206, 160); // Blue
ellipse(57, 79, 64, 64);
// Reas e Fry (2007, p88, 9-09)
```

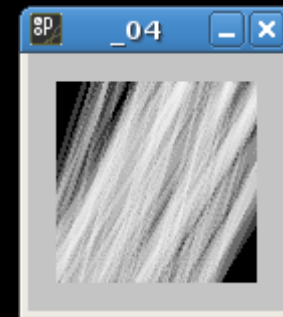


Um primeiro modo (mais simples, mas nem tanto)

```
// tipografia
PFont font;
font = loadFont("Ziggurat-72.vlw");
textFont(font);
fill(0, 80); // Black with low opacity
text("F", 0, 50);
text("I", 15, 65);
text("S", 30, 80);
text("L", 45, 95);
text("9", 60, 105);
// Reas e Fry (2007, p113, 13-05)
```



```
// matemática
background(0);
stroke(255, 60);
for (int i = 0; i < 100; i++) {
  float r = random(10);
  strokeWeight(r);
  float offset = r * 5.0;
  line(i-20, 100, i+offset, 0);
}
// Reas e Fry (2007, p129, 15-04)
```

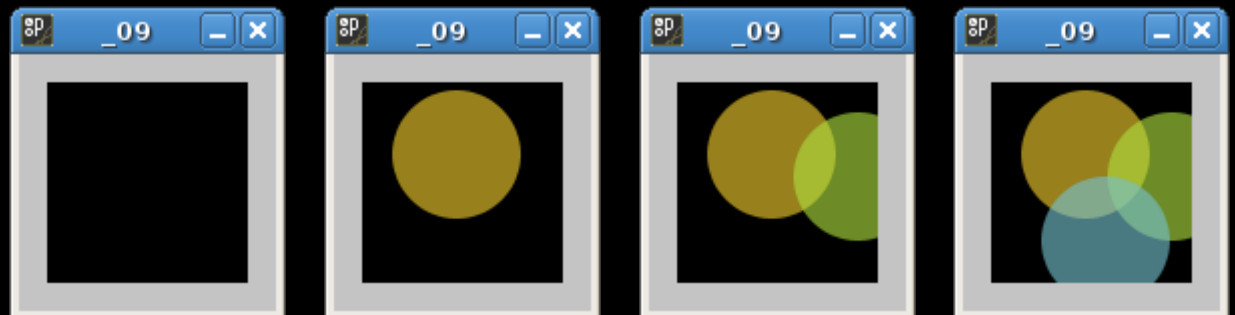


Utilizando Processing

Um primeiro modo (mais simples, mas nem tanto)

```
// cores
background(0);
noStroke();
smooth();
fill(242, 204, 47, 160);
ellipse(47, 36, 64, 64);
fill(174, 221, 60, 160);
ellipse(90, 47, 64, 64);
fill(116, 193, 206, 160);
ellipse(57, 79, 64, 64);
// Reas e Fry (2007, p88, 9-09)
```

```
// desenhar fundo em preto
// instrui para desenhar sem traço
// instrui para desenhar sem serrilhado (antialiasing)
// instrui para desenhar em um amarelo semi transparente
// instrui para desenhar uma elipse
// instrui para desenhar em um verde semi transparente
// instrui para desenhar uma segunda elipse
// instrui para desenhar em um azul semi transparente
// instrui para desenhar uma terceira elipse
```



Utilizando Processing

Um primeiro modo (mais simples, mas nem tanto)

// tipografia

PFont font;

font = loadFont("Ziggurat-72.vlw");

textFont(font);

fill(0, 80);

text("F", 0, 50);

text("I", 15, 65);

text("S", 30, 80);

text("L", 45, 95);

text("9", 60, 105);

// Reas e Fry (2007, p113, 13-05)

// Cria uma variável para trabalhar com texto do tipo Pfont

// Carrega uma fonte na memória

// Instrui para se escrever com a fonte recém carregada

// Instrui para se desenhar em preto opaco, semi-transparente

// Escrever o texto "F"

// Escrever o texto "I"

// Escrever o texto "S"

// Escrever o texto "L"

// Escrever o texto "9"



Utilizando Processing

Um primeiro modo (mais simples, mas nem tanto)

```
// matemática
size (800,100);
background(0);
stroke(255, 60);

for (int i = 0; i < 800; i++)
{
  float r = random(10);
  strokeWeight(r);
  float offset = r * 5.0;
  line(i-20, 100, i+offset, 0);
}
// baseado em Reas e Fry (2007, p129, 15-04)
```

// abrir tela com 800 x 100
// desenhar fundo em preto
// desenhar traços em branco opaco

// repetir um conjunto de instruções 800 vezes

// criar uma variável float r e atribuir um número qualquer até 10
// ajustar a espessura do traço
// criar uma segunda variável relacionada a inclinação
// desenhar uma linha de (i-20, 100) a (i+offset, 0)
// (inha de baixo para a linha de cima)

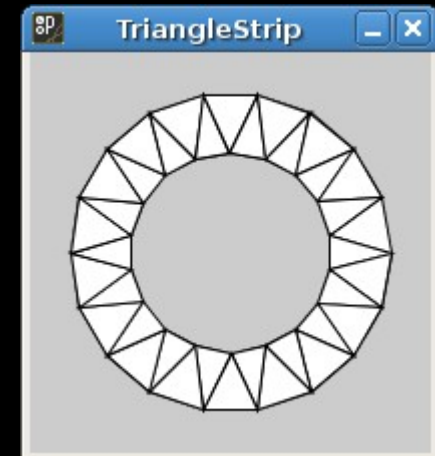


Utilizando Processing

```
/**
 * TRIANGLE_STRIP Mode
 * By Ira Greenberg
 * Generate a closed ring using vertex()
 * function and beginShape(TRIANGLE_STRIP)
 * mode. outerRad and innerRad variables
 * control ring's outer/inner radii respectively.
 * trig functions generate ring.
 * Examples -> Basics -> Form -> TriangleStrip
 */
size(200, 200);
background(204);
smooth();
int x = width/2;
int y = height/2;
int outerRad = 80;
int innerRad = 50;
float px = 0, py = 0, angle = 0;
float pts = 36;
float rot = 360.0/pts;
```

Um primeiro modo (mais simples, mas nem tanto)

```
beginShape(TRIANGLE_STRIP);
for (int i=0; i<pts; i++) {
  px = x+cos(radians(angle))*outerRad;
  py = y+sin(radians(angle))*outerRad;
  angle+=rot;
  vertex(px, py);
  px = x+cos(radians(angle))*innerRad;
  py = y+sin(radians(angle))*innerRad;
  vertex(px, py);
  angle+=rot;
}
endShape();
```



Em seu livro, **Reas e Fry (2007, p. 1-7)** caracterizam Processing em relação a:

1. Software:

- Software é um meio único com qualidades únicas [que se diferencia por] sua habilidade em produzir formas dinâmicas, processar gestos, definir comportamentos, simular sistemas naturais e integrar outros medias incluindo som, imagem e texto (REAS e FRY, 2007, p.1).
- Toda linguagem de programação é um material distinto (id. p. 1-2);
- Esboçar é necessário para o desenvolvimento das idéias (ibidem, p. 3);
- Programação não é apenas para engenheiros[as] (ibidem, p. 3);

2. Letramento:

Processing não apresenta uma mudança radical da atual cultura de programação.

Mas reposiciona tal atividade de uma forma acessível à pessoas que estão interessadas em programação mas que podem se sentir intimidadas ou desinteressadas pelas maneiras comumente ensinadas em departamentos de Ciência da Computação (REAS e FRY, 2007, p.3).

Em seu livro, **Reas e Fry (2007, p. 1-7)** caracterizam Processing em relação a:

3. Abertura:

O movimento de **código aberto** vem tendo um grande impacto na nossa cultura e economia devido a iniciativas como o Linux, mas ainda vem tendo uma influencia pequena na cultura em torno de software para as artes.

(ibidem, p. 4)

4. Educação;

Processing torna possível tanto **introduzir conceitos de software no contexto das artes** como **levar conceitos das artes a audiências mais técnicas** (p. 4);

Como a linguagem em processing é **derivada de linguagens vastamente utilizadas**, representa uma boa base para o futuro aprendizado;

Ensinar técnicas básicas de programação e simultâneamente introduzir teorias básicas propicia aos/às estudantes a exploração de suas idéias diretamente, assim como o desenvolvimento de uma compreensão profunda e uma intuição sobre mídias digitais e interativas.

Em seu livro, **Reas e Fry (2007, p. 1-7)** caracterizam Processing em relação a:
5. Rede;

Programas em processing são **facilmente exportáveis para a web**, provendo suporte para colaboração pela rede e para compartilhamento de trabalhos] desenvolvidos (ibidem p. 6)

6. Contexto:

A abordagem à programação se mescla com métodos estabelecidos.

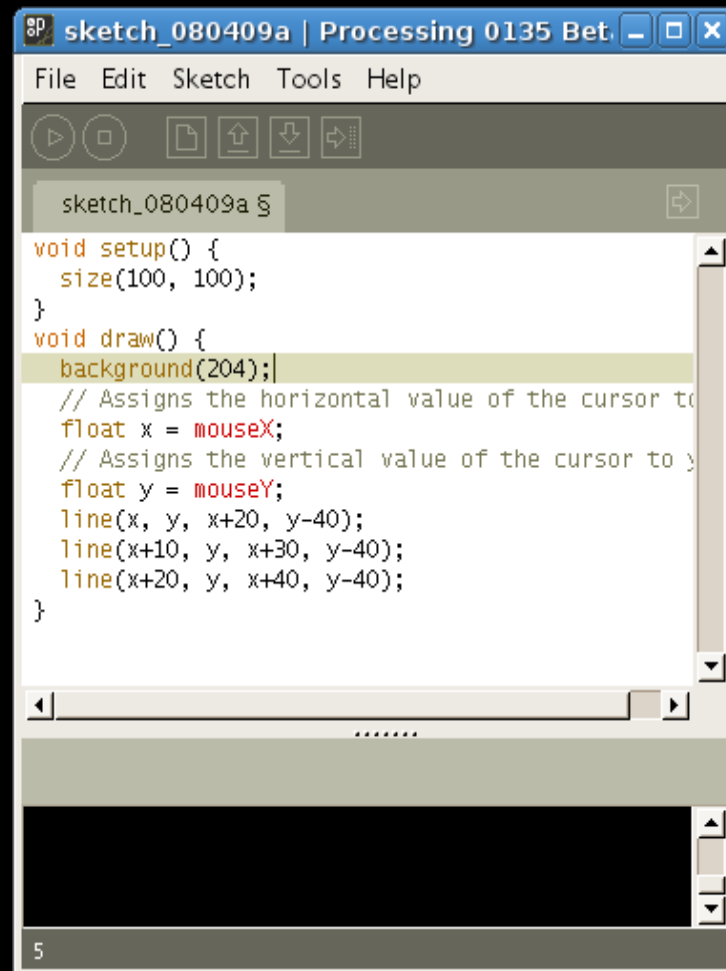
O núcleo da linguagem e as bibliotecas adicionais fazem uso de Java, que apresenta elementos semelhantes à linguagem de programação C. (ibidem p. 7)

Utilizando Processing

Um segundo modo (potencialmente **procedural** e **estruturado**)

Utilizando Processing

```
void setup() {  
  size(100, 100);  
}
```



Um segundo modo (mais comum)

```
// Desenhar  
void draw() {  
  background(204);  
  // Atribui o valor horizontal  
  // do cursor à variável x  
  float x = mouseX;  
  // Atribui o valor vertical  
  // do cursor à variável y  
  float y = mouseY;  
  line(x, y, x+20, y-40);  
  line(x+10, y, x+30, y-40);  
  line(x+20, y, x+40, y-40);  
}  
//Reas e Fry (2007, p. 13, 0-05)
```



dependendo da posição do mouse

Utilizando Processing

Um segundo modo (procedural + aberto)

O segundo modo **mais simples** de se programar em processing permite o desenvolvimento de esboços mais complexos, que também podem envolver:

- o redesenho de uma forma em outro lugar, podendo envolver **animação**;
- a estruturação de códigos mais longo por meio de **funções**;
- o uso mais extenso de **bibliotecas**, inclusive **gráficas**, incluindo o desenho em ambientes bidimensionais e tridimensionais (cor, transformações, etc.); (Isto também pode ser feito no primeiro modo)
- **interatividade**, por meio da programação de **eventos** (teclado/mouse);

Junto aos esboços mais complexos envolvendo funções, é comum se encontrar usos de classes, principalmente naqueles em simulação e em animação.

Entretanto, o desenvolvimento orientado a objetos (terceiro modo) não é uma exigência, mas um recurso. O segundo modo envolve:

uma preparação **void setup(){} e um laço a redesenhar void draw(){} e**

Utilizando Processing

Um segundo modo (mais comum)

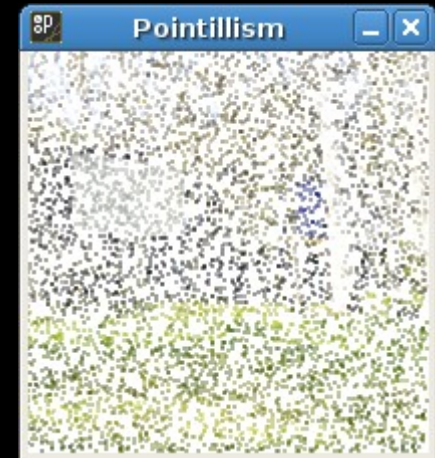
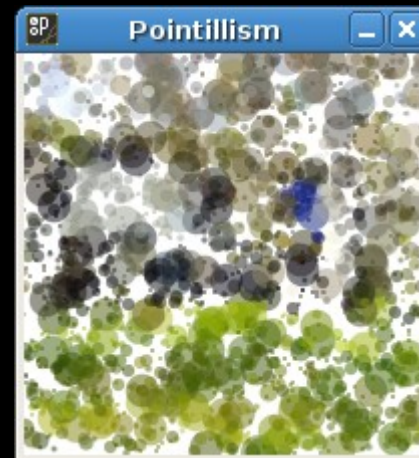
```
/**
 * Pointillism
 * by Daniel Shiffman.
 * Mouse horizontal location controls size of dots.
 * Creates a simple pointillist effect using ellipses
 * colored according to pixels in an image.
 * Created 2 May 2005
 */
```

```
PImage a;
void setup()
{
  a = loadImage("eames.jpg");
  size(200,200);
  noStroke();
  background(255);
  smooth();
}
```



```
void draw()
{
  float pointillize = map(mouseX, 0, width, 2, 18);
  int x = int(random(a.width));
  int y = int(random(a.height));
  color pix = a.get(x, y);
  fill(pix, 126);
  ellipse(x, y, pointillize, pointillize);
}
```

dependendo do mouse



Utilizando Processing

Um segundo modo (mais comum)

```
/**
 * Pointillism
 * by Daniel Shiffman.
 * Mouse horizontal location controls size of dots.
 * Creates a simple pointillist effect using ellipses
 * colored according to pixels in an image.
 * Created 2 May 2005
 */
```

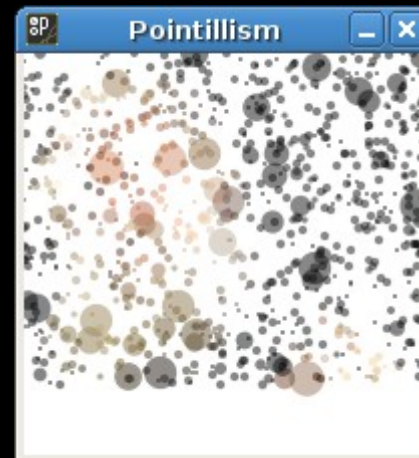
```
PImage a;
void setup()
{
  a = loadImage("maddoghall.jpg");
  size(200,200);
  noStroke();
  background(255);
  smooth();
}
```



Jon "Maddog" Hall

```
void draw()
{
  float pointillize = map(mouseX, 0, width, 2, 18);
  int x = int(random(a.width));
  int y = int(random(a.height));
  color pix = a.get(x, y);
  fill(pix, 126);
  ellipse(x, y, pointillize, pointillize);
}
```

dependendo do mouse e da imagem



Paulo Freire (1996) **Pedagogia da Autonomia:**
Saberes necessários à prática educativa,
Paz e Terra

Capítulo I:
Não há docência sem discência

Ensinar exige rigorosidade metódica

Ensinar exige pesquisa

Ensinar exige respeito aos saberes dos educandos

Ensinar exige criticidade

Ensinar exige estética e ética

Ensinar exige a corporeificação das palavras pelo exemplo

Ensinar exige risco, aceitação do novo e rejeição
a qualquer forma de discriminação

Ensinar exige reflexão crítica sobre a prática

Ensinar exige o reconhecimento e
a assunção da identidade nacional

Capítulo II

Ensinar não é transferir conhecimento

Ensinar exige consciência do inacabamento

Ensinar exige o reconhecimento de ser condicionado

Ensinar exige respeito à autonomia do ser do educado

Ensinar exige bon senso

Ensinar exige humildade, tolerância e luta em defesa dos
direitos dos educadores

Ensinar exige apreensão da realidade

Ensinar exige alegria e esperança

Ensinar exige a convicção de que a mudança é possível

Ensinar exige curiosidade

Capítulo III

Ensinar é uma especificidade humana

Ensinar exige segurança, competência profissional
e generosidade

Ensinar exige comprometimento

Ensinar exige compreender que a educação é
uma forma de intervenção no mundo

Ensinar exige liberdade e autoridade

Ensinar exige tomada consciente de decisões

Ensinar exige saber escutar

Ensinar exige reconhecer que a educação é ideológica

Ensinar exige disponibilidade para o diálogo

Ensinar exige querer bem aos educandos

Utilizando Processing

Um terceiro modo (potencialmente **orientado a objetos**)

Utilizando Processing

Um terceiro modo (orientado a objetos)

Um terceiro modo de se programar em processing envolve o uso de recursos orientados a objetos. Isto permite que:

- se aproxime as linguagens utilizadas para desenvolver o esboço e para representar o domínio deste; Se diminua o gap semântico da linguagem;
- se eleve o nível de abstração do programa;
- se reutilize código com mais eficiência;

Utilizando Processing

//Reas e Fry (2007)

Diagonals da, db;

```
void setup() {  
  size(100, 100);  
  smooth();  
  // Entradas: x, y, velocidade, espessura, cinza  
  da = new Diagonals(0, 80, 1, 2, 0);  
  db = new Diagonals(0, 55, 2, 6, 255);  
}
```

```
void draw() {  
  background(204);  
  da.update();  
  db.update();  
}
```

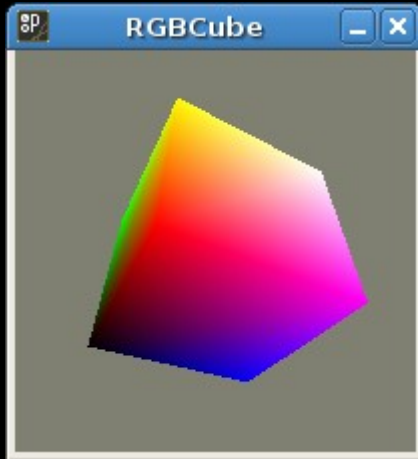
Um terceiro modo (orientado a objetos)

```
class Diagonals {  
  int x, y, speed, thick, gray;  
  Diagonals(int xpos, int ypos, int s, int t, int g) {  
    x = xpos;  
    y = ypos;  
    speed = s;  
    thick = t;  
    gray = g;  
  }  
  void update() {  
    strokeWeight(thick);  
    stroke(gray);  
    line(x, y, x+20, y-40);  
    line(x+10, y, x+30, y-40);  
    line(x+20, y, x+40, y-40);  
    x = x + speed;  
    if (x > 100) {  
      x = -100;  
    }  
  }  
}
```

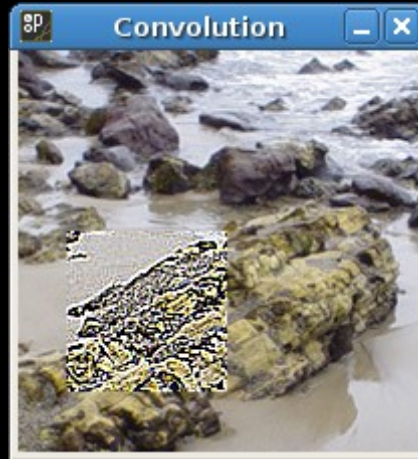


Utilizando Processing

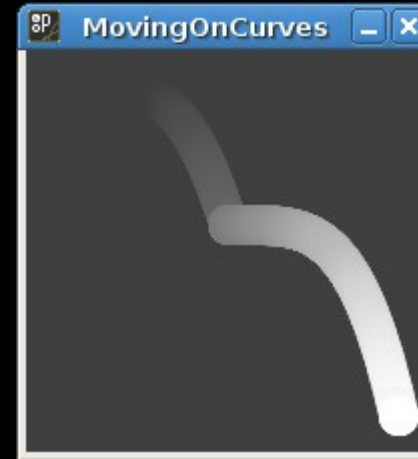
Tópicos / Exemplos (Ambiente)



3D->Form-> **RGBCube**



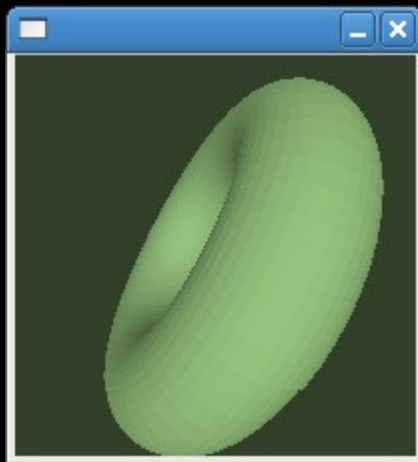
Topics -> Image P. -> **Convolution**



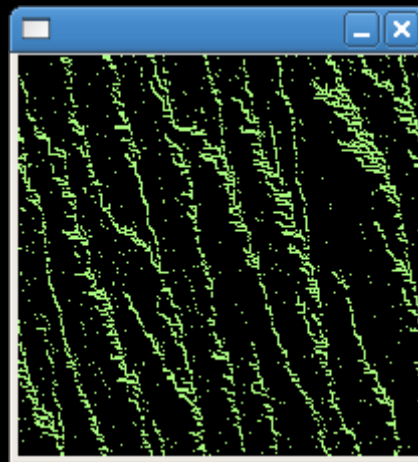
Topics -> Motion -> **Moving OC**



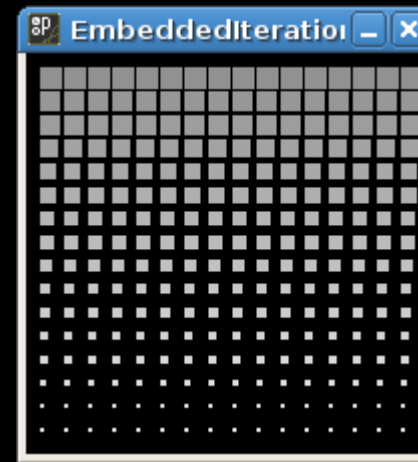
Topics -> Simulate -> **Fluid**



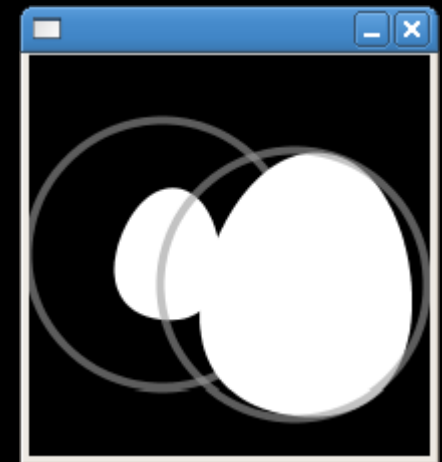
3D->Form-> **Toroid**



Topics -> Cel. Aut. -> **Spore1**



Basics -> Control -> **Emb. Interaction**



Basics -> Objects -> **CompositeO**

Utilizando Processing

Outros exemplos (Livro Reas e Fry (2007))



Utilizando Processing

Exemplos Básicos (Ambiente)

Structure

- Statements and ...
- Coordinates
- Width and Height
- Setup and Draw
- No Loop
- Loop
- Redraw
- Functions
- Recursion
- Recursion 2
- CreateGraphics

Form

- Points and Lines
- Shape Primitives
- Simple Curves
- Pie Chart
- Vertices
- Triangle Strip
- Bezier
- Bezier Ellipse

Data

- Variables
- Integers and Floats
- True/False
- Characters and Strings
- Datatype Conversion
- Variable Scope

Control

- Iteration
- Embedded Iteration
- Conditionals 1
- Conditionals 2
- Logical Operators

Image

- Displaying
- Background Image
- Pointillism
- Transparency
- Sprite
- Alphamask
- CreateImage

Math

- Increment/Decrement
- Operator Precedence
- Modulo
- Distance 1D
- Distance 2D
- Sine
- Sine and Cosine
- Sine Wave
- Additive Wave
- Polar to Cartesian
- Arctangent
- Graphing 2D Equation
- Random
- Double Random
- Noise 1D
- Noise 2D
- Noise 3D
- NoiseWave

Transform

- Translate
- Scale
- Rotate
- Triangle Flower
- Arm

Color

- Hue
- Saturation
- Brightness
- Color Wheel
- Reading
- Creating
- Relativity
- Linear Gradient
- Radial Gradient
- Wave Gradient

Input

- Mouse 1D
- Mouse 2D
- MousePress
- Mouse Signals
- Easing
- Constrain
- Storing Input
- Mouse Functions
- Keyboard
- Keyboard Functions
- Milliseconds
- Clock

Arrays

- Array
- Array 2D
- Array Objects

Objects

- Objects
- Multiple Constructors
- Composite Objects
- Inheritance
- Neighborhood

Typography

- Letters
- Words

Web

- Embedded Links
- Loading Images

Utilizando Processing

Tópicos / 3D e OpenGL (Ambiente)

Tópicos

Animação
Automatos Celulares
Desenho
E/S Arquivos
Fractais e Sistemas-L
GUI
Processamento de Imagem
Interação
Movimento
Simulação

3D e OpenGL

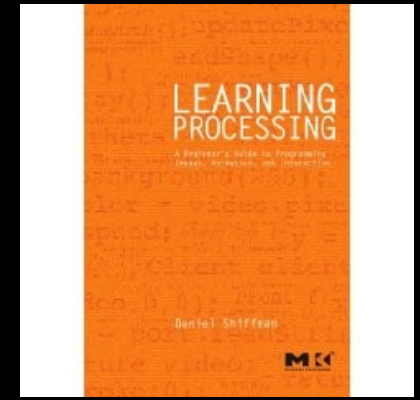
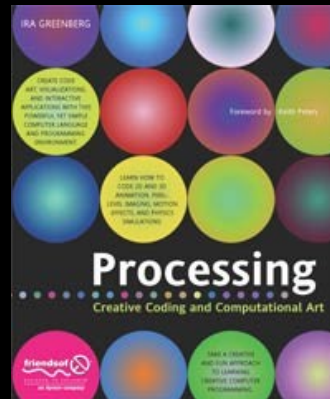
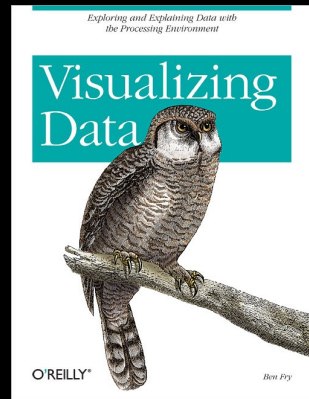
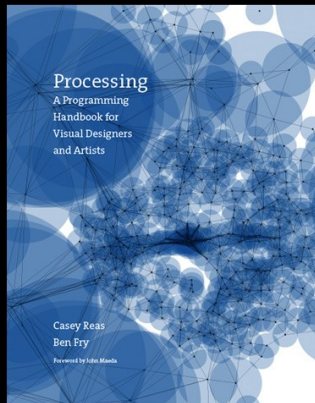
Forma
Transformação
Imagem
Tipografia
Iluminação
Câmera
Texturas
OpenGL

Bibliotecas

Video (Captura)
Video (Filme)
Candy (SVG Import)
PDF (Export)
Rede
Serial
Ess (Som)

Outras

Livros e Monografias



Casey Reas and Ben Fry Processing: A Programming Handbook for Visual Designers and Artists (Foreword by John Maeda). Published 24 August 2007, MIT Press. 736 pages.

Ben Fry. Visualizing Data Published December 2007, O'Reilly. 384 pages. Paperback.

Ben Fry. Computational Information Design. PhD Thesis, Massachusetts Institute of Technology. Program in Media Arts and Sciences, 2004. Disponível em <http://benfry.com/phd/dissertation-050312b-acrobat.pdf>. Acesso em 11/04/2008.

Ira Greenberg Processing: Creative Coding and Computational Art Published 28 May 2007, Friends of Ed. 840 pages.

Tom Igoe. Making Things Talk: Practical Methods for Connecting Physical Objects Published 28 September 2007, O'Reilly. 428 pages. Paperback.

Daniel Shiffman Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction (Morgan Kaufmann Series in Computer Graphics)

Yguá
PPGTE
merkle@utfpr.edu.br
obrigado !