

# 2

## Autoria Multimédia

## Autoria Multimédia

- Desenvolvimento de aplicações multimédia, combinando vários media no espaço e no tempo.
  - Habitualmente interactivo
- Suportado por ferramentas (*authoring tools*), linguagens de script e prototipagem
- Normalmente uma das fases de desenvolvimento de um projecto, desde o planeamento até à entrega aos utilizadores finais

## Projecto Multimédia

1. **Análise e planeamento:** Objectivos. Ideias e mensagens. Tempo. Capacidades e recursos . Custos. Plano.
2. **Design:** Que tipos de *media*. Combinados de que forma. Que tipo de navegação => guião da aplicação e protótipo.
3. **Produção:** Criação de conteúdos e aplicações com ferramentas de autoria
4. **Teste e validação:** detecção e correcção de erros e inconsistência. Objectivos atingidos? Aplicação corre nas plataformas pretendidas? **Normalmente não...**
5. **Entrega e manutenção:** Produto final entregue e mantido nas plataformas pretendidas (offline ou online).

PCM (DI/FCT/UNL) ©nmc 2012

## Projecto Multimédia

- **Gestor de projecto.**
- **Designer multimédia.**  
por exemplo: d. gráfico, animador, *instrucional* d., d. de informação, d. de interfaces.
- **Writer. Especialista em vídeo. Especialista em áudio.**
- **Programador multimédia.**
- **Produtor multimédia/Produtor Web**

Os papéis podem ser desempenhados pela mesma pessoa ou por várias.

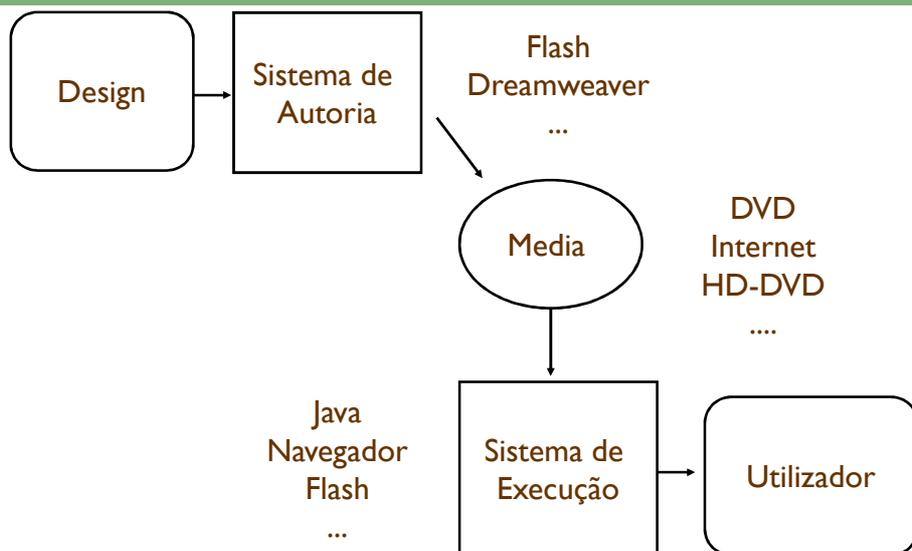
PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria

- Programas para o desenvolvimento de títulos multimédia interactivos
  - São essencialmente ferramentas de integração de diferentes materiais (texto, imagem, áudio, vídeo)
- Combinam conceitos das linguagens tradicionais com ambientes visuais de programação
- Por vezes combinam autoria de conteúdos (imagens, vídeo), autoria multimédia e scripting
- Prototipagem rápida
- Aceleram o processo de desenvolvimento

PCM (DI/FCT/UNL) ©nmc 2012

## Processo de Desenvolvimento



PCM (DI/FCT/UNL) ©nmc 2012

## Projecto Multimédia

- **Tem por objectivos definir e criar:**

- Mensagem
- Conteúdo
- Navegação
- Interface
- Aplicação

- **Interactividade:** nem sempre presente

- e.g., *slide show* ou filme

---

PCM (DI/FCT/UNL) ©nmc 2012

## Metáforas/Paradigmas

Define os elementos e a metodologia usados na construção dos títulos. Alguns exemplos:

- Linguagens de script
- Slide Show (PowerPoint)
- Ícones/Controlo de fluxo
- Cartões/Scripting/Livro
- Tempo/Markup
- Tempo/Pauta/Teatro/Cinema
- ...combinações dos anteriores

---

PCM (DI/FCT/UNL) ©nmc 2012

## Linguagens de Script

- É o paradigma mais semelhante às linguagens tradicionais
- Especificação textual dos elementos multimédia, sequência, sincronização, etc.
- Baseadas num modelo de objectos, com eventos/respostas a eventos e mensagens
- Normalmente interpretadas e por isso não implica necessariamente mais rapidez relativamente aos outros paradigmas
- Exemplos: Javascript, ActionScript

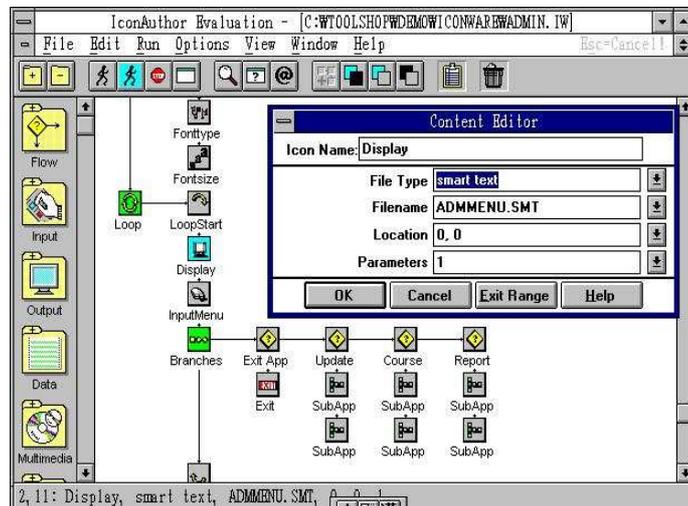
*PCM (DI/FCT/UNL) ©nmc 2012*

## Ícones/Controlo de fluxo

- Menu de objectos e de funções do programa
- Representados no espaço de trabalho como ícones
- Processo de desenvolvimento mais rápido
- Execução mais lenta pois cada primitiva de programação é avaliada para todas as possibilidades (optimizado nalguns casos)
- Exemplos: IconAuthor, MAX/MSP

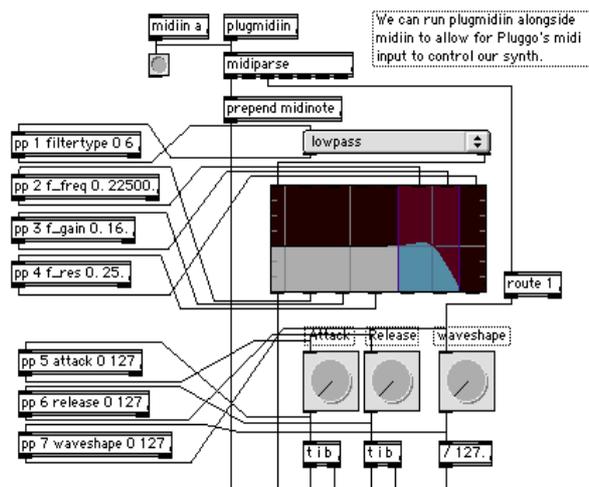
*PCM (DI/FCT/UNL) ©nmc 2012*

# IconAuthor



PCM (DI/FCT/UNL) ©nmc 2012

# MAX/MSP



PCM (DI/FCT/UNL) ©nmc 2012

## Cartões/Scripting/Livro

- Estrutura bem definida, mas por vezes demasiado rígida:
  - Cartões
  - Livro
- Deriva dos modelos/aplicações de hipertexto
- Normalmente extensível através de DLLs ou mecanismos semelhantes
- A combinação com as linguagens de *scripting* torna-os bastantes eficazes para prototipagem
- Exemplos: HyperCard, ToolBook

PCM (DI/FCT/UNL) ©nmc 2012

## Tempo/Markup

- Estrutura temporal especificada numa linguagem de *markup*, inclui a estrutura do documento e a sincronização
- Aproximação declarativa vs. procedimental
- Autoria usando um editor de texto ou uma ferramenta específica
- Por exemplo, SMIL – “Synchronized Multimedia Integration Language” uma recomendação internacional W3C para a Web
- Permite controlar: “o quê”, “onde”, “quando” dos elementos numa apresentação multimédia: sincronização e posicionamento de objectos multimédia

PCM (DI/FCT/UNL) ©nmc 2012

## Tempo/Markup

### Estrutura de um documento SMIL

```
<?xml version="1.0"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
    "http://www.w3.org/2001/SMIL20/SMIL20.dtd">

<smil xmlns="http://www.w3.org/2001/SMIL20/Language"
    xml:lang="en" title="SMIL template">

<head>
</head>
<body>
</body>

</smil>
```

PCM (DI/FCT/UNL) ©nmc 2012

## Tempo/Markup

### Exemplo: Sincronização

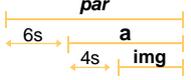
**seq** Os filhos do elemento seq são sequenciais no tempo

```
<seq>
  <audio src="audio1" />
  <audio begin="5s" src="audio2" />
</seq>
```



**par** Os filhos do elemento par podem sobrepôr-se no tempo

```
<par>
  <audio id="a" begin="6s" ... />
  <img begin="id(a) (4s)" ... />
</par>
```



PCM (DI/FCT/UNL) ©nmc 2012

## Pauta/Teatro/Cinema

- Os elementos sincronizados são alinhados em diversas pistas (*tracks*) horizontais
- Utilizados para animação ou outras aplicações com requisitos temporais
- Comportamento de cada objecto programado com linguagens de *script*
- Exemplo: Flash

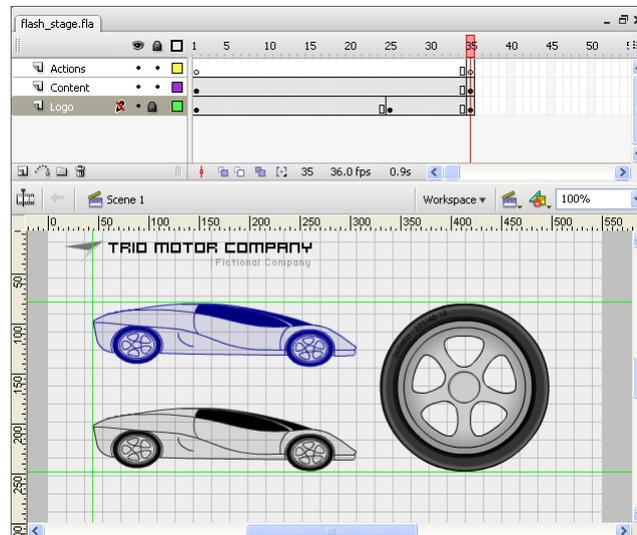
PCM (DI/FCT/UNL) ©nmc 2012

## Flash

- Flash é um programa para criar filmes interactivos ou aplicações.
- Visualizados na Web ou *players*
- Usa gráficos vectoriais e bitmap, ...
- Ambiente de autoria e *player*, que suporta todo o processo desde a produção de conteúdos à codificação
- O Flash suporta desenhos e animação vectorial, o que permite alterar dimensões sem perda de qualidade.
- Inclui uma linguagem designada por ActionScript que permite programar a componente interactiva

PCM (DI/FCT/UNL) ©nmc 2012

# Flash



PCM (DI/FCT/UNL) ©nmc 2012

# Flash

- Muitos dos websites com animação, áudio e interacção é feita em Flash
- Pode ser apenas parte da página (anúncios ou menus, por exemplo)
- Cerca de 98% dos computadores têm Flash instalado (Player)
- Exemplo:
  - <http://www.youtube.com>

PCM (DI/FCT/UNL) ©nmc 2012

## Flash

- É possível construir páginas inteiras em Flash. No entanto há alguns problemas nesta aproximação:
  - Indexação pelos search engines
  - Dificuldade em fazer bookmarks
  - Menos acessível para utilizadores cegos ou com dificuldades visuais (síntese de voz).
  - Funcionalidade habitual de botão direito desactivada.
  - Copy/paste muitas vezes não é possível.
- Normalmente uma combinação de HTML e Flash funciona melhor.
- Actualmente RIA (Rich Internet Applications)...

*PCM (DI/FCT/UNL) ©nmc 2012*

## Aplicações

- O Flash pode ser usado para criar aplicações que não necessitam de um Player.
- Usado para distribuição offline (CD, DVD).
- Usado para prototipagem rápida.
- Equipas multidisciplinares que envolvem programadores, artistas e designers.

*PCM (DI/FCT/UNL) ©nmc 2012*

## Publicação em Flash

Quando o filme está pronto pode ser publicado como:

- Um ficheiro SWF (ShockWave)
  - Necessita de um Player de Flash.
- Uma página HTML
  - Pode ser visualizada num Browser.
- Um ficheiro "projector"
  - Pode funcionar em modo standalone. Não necessita de ter o Flash instalado.
- Opções disponíveis a partir do menu "Publish".

*PCM (DI/FCT/UNL) ©nmc 2012*

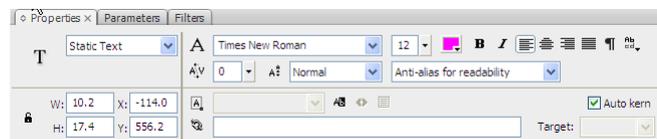
## Funcionalidades

- O Flash contém diversas ferramentas para criar conteúdo interactivo
- É possível desenhar, animar, codificar e finalmente exportar um filme que pode ser usado num Website ou numa aplicação.
- Elementos da interface:
  - Stage
  - Timeline and layers
  - Library
  - Drawing Tools
  - Symbols
  - Import from other programs

*PCM (DI/FCT/UNL) ©nmc 2012*

## Stage

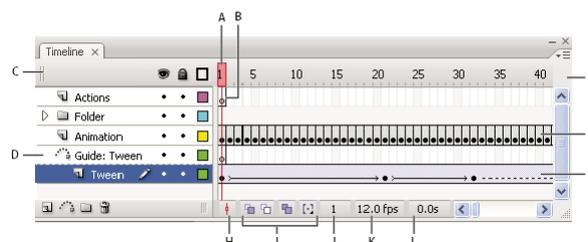
- É no Stage que se define o comportamento espacial do filme e se colocam os elementos gráficos ou multimédia.
- É uma área de desenho, com propriedades (Properties) que permitem mudar o tamanho, background e outras características.



PCM (DI/FCT/UNL) ©nmc 2012

## Timeline e Layers

- A Timeline permite animar e definir o comportamento dos diversos elementos.
- Suporta o conceito de Layer, tal como no PhotoShop e outros programas gráficos e de imagem.



PCM (DI/FCT/UNL) ©nmc 2012

## Library

- A Library contém símbolos (Symbols) que são usados no filme.
- Os símbolos podem ser imagens, animações, vídeos e outros elementos.
- Os símbolos podem ser arrastados para o Stage ou usados em animações na Timeline.
- Os símbolos no Stage chamam-se Movie Clips.

*PCM (DI/FCT/UNL) ©nmc 2012*

## HTML5

- Interacção na Web
- Integração de várias linguagens:
  - CSS
  - SVG (Standard Vector Graphics)
  - MathML
  - Video, Canvas, Geolocation, ...
- HTML e XML
  - Integra SVG e MathML
  - Integra DOM Core e DOM HTML

*PCM (DI/FCT/UNL) ©nmc 2012*

## HTML5

- SVG (Scalable Vector Graphics)
- SVG 1.1 lançado em 2001 e SVG 1.2 lançado em 2009
- SVG 1.1 incluído no HTML 5
- Opera suporta SVG 1.1/1.2
- Firefox e Webkit suportam grande parte do SVG 1.1
- Microsoft participa no SVG Working Group desde Janeiro de 2010

*PCM (DI/FCT/UNL) ©nmc 2012*

## HTML5

- Novos objectos (Video), H263, H264, ...

```
<video src='myMovie' id='vid' />
var vid = document.getElementById("vid");
vid.play();
vid.pause();
vid.currentTime = 0;
```

*PCM (DI/FCT/UNL) ©nmc 2012*

## HTML5 (SVG)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">

<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>

</svg>
```

PCM (DI/FCT/UNL) ©nmc 2012

## HTML5 (Canvas Interface)

```
interface CanvasRenderingContext2D {
    [...]
    void scale(in float x, in float y);
    void rotate(in float angle);
    void translate(in float x, in float y);
    CanvasGradient createLinearGradient(...);
    CanvasPattern createPattern(in HTMLElement element,
                               in DOMString repetition);
    void strokeRect(in float x, in float y, in float w, in
float h);
    void moveTo(in float x, in float y);
    void strokeText(...);
};
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing

- Projecto open source project iniciado por Casey Reas and Benjamin Fry, quando estavam no Aesthetics and Computation Group do MIT Media Lab.
- Linguagem de programação e ambiente de desenvolvimento
- Construído a partir das características da linguagem Java, simplificando algumas características e adicionando novas.

*PCM (DI/FCT/UNL) ©nmc 2012*

## Processing

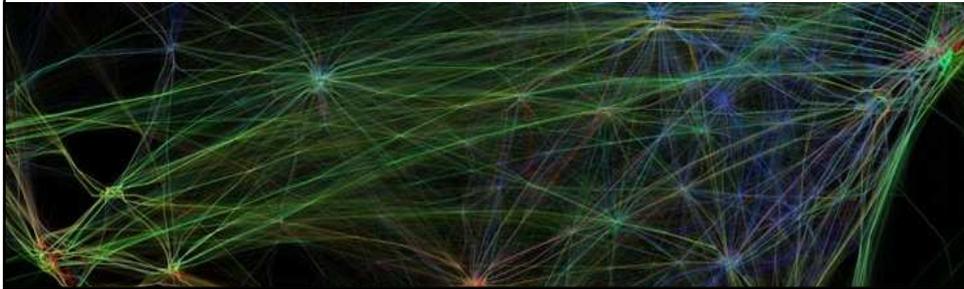
### Processing:

- O ambiente de desenvolvimento ou Processing Development Environment (PDE).
- Uma colecção de funções ou métodos que são o core da interface de programação ou API.
- Várias bibliotecas que suportam funcionalidades avançadas tal como desenhar com OpenGL ou ler ficheiros XML
- Uma sintaxe, semelhante à da linguagem Java.

*PCM (DI/FCT/UNL) ©nmc 2012*

## Processing – Exemplo (1)

- Usado em visualização e instalações interactivas
- Exemplo: Rotas aéreas sobre a América do Norte visualizadas em cor e forma (Aaron Koblin)



## Processing

### Sketches (programas) estáticos

```
// Versão 1
line(15, 25, 70, 90);

// Versão 2
size(400, 400);
background(192, 64, 0);
stroke(255);
line(150, 25, 270, 350);
Print("Hello World");
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - Métodos

- **setup** é executado uma vez e inclui habitualmente inicializações

```
void setup() {  
    size(400, 400);  
    stroke(0);  
    background(255, 0, 0);  
    frameRate(30);  
}
```

- **draw** é executado repetidamente (animação)

```
void draw() {  
    // quantas linhas estão visíveis em cada momento?  
    line(100, 50, mouseX, mouseY);  
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - OOP

```
float x = 33;  
float y = 50;  
float diameter = 30;
```

```
void setup() {  
    size(100, 100);  
    smooth();  
    noStroke();  
}  
void draw() {  
    background(0);  
    ellipse(x, y, diameter, diameter);  
}
```



PCM (DI/FCT/UNL) ©nmc 2012

# Processing - OOP

```
Spot sp;
void setup() {
  size(100, 100);
  smooth();
  noStroke();
  sp = new Spot();
  sp.x = 33;
  sp.y = 50;
  sp.diameter = 30;
}
void draw() {
  background(0);
  ellipse(sp.x, sp.y, sp.diameter,
    sp.diameter);
}
```

Classe Spot



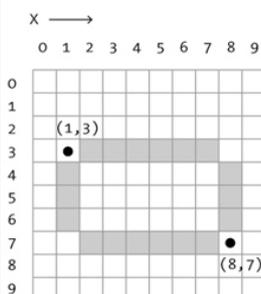
```
class Spot {
  float x, y; // The x- and y-
  coordinate
  float diameter; // Diameter of the
  circle
}
```

Outros métodos? Construtor, display() semelhante ao draw

PCM (DI/FCT/UNL) ©nmc 2012

# Processing - Gráficos

Várias primitivas: point, line, rect, ellipse, bezier, curve, vertex, curveVertex, bezierVertex



```
ellipseMode(CORNERS);
ellipse(x1,y1,x2,y2);
```

Example:  
 ellipseMode(CORNERS);  
 ellipse(1,3,8,7);



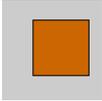
```
noFill();
beginShape();
curveVertex(84, 91);
curveVertex(84, 91);
curveVertex(68, 19);
curveVertex(21, 17);
curveVertex(32, 100);
curveVertex(32, 100);
endShape();
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - Gráficos

Método fill define o que acontece às formas fechadas. Várias formas de especificar:

```
fill(gray)
fill(gray, alpha)
fill(value1, value2, value3)
fill(value1, value2, value3, alpha)
fill(color)
fill(color, alpha)
fill(hex)
fill(hex, alpha)
```



```
fill(204, 102, 0);
rect(30, 20, 55, 55);

// ou
color c = color(200, 0, 0);
fill(c);
Rect(30,20,55,55);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - Interacção

Mouse: mouseX, mouseY, pmouseX, pmouseY,  
mousePressed(), mouseReleased(), mouse Dragged(),  
mouseButton, cursor(), noCursor()

```
// Exemplo 1
void draw() {
background(255);
ellipse(mouseX, mouseY, 100, 100); // coordenadas actualizadas
do rato
}

// Exemplo 2 - mousePressed, executado ao clicar
void mousePressed() {
if (mouseButton == LEFT)
print(mouseX + "\n");
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - Interacção

Teclado: keyPressed é executado quando é premida uma tecla. Key indica qual a tecla.

```
void keyPressed(){
  print("The key is "+key);
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing - Interacção

pmouseX, pmouseY guardam os valores da *frame* anterior

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

void draw() {
  background(204);
  line(mouseX, mouseY, pmouseX, pmouseY);
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Animação

Técnica de Easing (Interpolação)

Usada em animação tradicional, para fazer inícios e paragens suaves

```
float x, y, targetX, targetY;
float easing = 0.05;

void setup()
{
  size(400, 400);
  smooth();
  noStroke();
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Animação

■ Técnica de Easing (Interpolação)

```
void draw()
{
  background( 51 );

  targetX = mouseX;
  float dx = targetX - x;
  if(abs(dx) > 1) {
    x += dx * easing;
  }
  // same for y

  ellipse(x, y, 66, 66);
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Imagens representadas pela classe Pimage, loadImage para carregar, image para mostrar e tint/notint para mudar a cor:

```
PImage img;
void setup() {
  size(320,240);
  img = loadImage("mypic.jpg");
}
void draw() {
  background(0);
  tint(0, 153, 204); // Blue
  image(img,0,0);
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Pixels acessados por get/set (da janela ou de uma imagem):

```
// Example 1: single pixel
PImage img = loadImage("tower.jpg");
background(img);
noStroke();
color c = img.get(60, 90);
fill(c);
rect(25, 25, 50, 50);

// Example 2: group of pixels
PImage img = loadImage("tower.jpg");
background(img);
PImage b = img.get(50, 0, 50, 100);
image(b, 0, 0);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Pixels acedidos por get/set (da janela ou de uma imagem):

```
// Example 3: pixel in image
PImage trees;
trees = loadImage("tree.jpg");
background(0);
color white = color(255);
trees.set(0, 50, white);
trees.set(1, 50, white);
trees.set(2, 50, white);
trees.set(3, 50, white);
image(trees, 20, 0);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Filtros, combinação e cópia de imagens

- filter, blend e copy
- filter(mode) ou filter(mode, level) – **mode** indica o tipo de filtro (THRESHOLD, GRAY, INVERT, POSTERIZE, BLUR, OPAQUE, ERODE, or DILATE) e **level** quando é aplicado o filtro (apenas nalguns casos)

```
PImage img = loadImage("pic.jpg");
img.filter(THRESHOLD, 0.3); // Acima de 30% do
    máximo os pixels ficam brancos. Abaixo pretos.
image(img, 0, 0);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Blend permite combinar duas imagens de acordo com o modo:

- `blend(x, y, width, height, dx, dy, dwidth, dheight, mode)`
- `blend(srcImg, x, y, width, height, dx, dy, dwidth, dheight, mode)`

```
PImage img = loadImage("forest.jpg");
PImage img2 = loadImage("airport.jpg");
img.blend(img2, 12, 12, 76, 76, 12, 12, 76, 76, ADD);
image(img, 0, 0);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Copy - permite copiar imagens:

- `copy(x, y, width, height, dx, dy, dwidth, dheight)`
- `copy(srcImg, x, y, width, height, dx, dy, dwidth, dheight)`

```
// Exemplo 1
PImage img = loadImage("forest.jpg");
image(img, 0, 0);
copy(0, 0, 100, 50, 0, 50, 100, 50);

// Exemplo 2
PImage img = loadImage("tower.jpg");
img.copy(50, 0, 50, 100, 0, 0, 50, 100);
image(img, 0, 0);
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Imagens

Processamento: acesso directo aos pixels

```
pixels[], loadPixels(), updatePixels()
// Negativo
PImage arch = loadImage("pic.jpg");
background(arch);
loadPixels();
for (int i = 0; i < width*height; i++) {
  color p = pixels[i];
  float r = 255 - red(p);
  float g = 255 - green(p);
  float b = 255 - blue(p);
  pixels[i] = color(r, g, b);}
updatePixels();
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Vídeo

Class Movie: Armazenamento e reprodução de filmes

```
import processing.video.*;
Movie myMovie;

void setup() {
  size(200, 200, P2D);
  myMovie = new Movie(this, "news.mov");
  myMovie.loop(); // reproduz o filme continuamente
}

void draw() {
  image(myMovie, mouseX, mouseY);
}

// Chamada cada vez que há uma nova imagem
void movieEvent(Movie m) {
  m.read();
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Vídeo

Guardar uma imagem (exemplo anterior)

```
Pimage myImage;

void movieEvent(Movie m) {
    m.read();
    myImage = m.get() ;
}

void keyPressed(){
    // no primeiro caso guarda frame da janela, no segundo directamente do
    // filme e no terceiro usa a classe Pimage que permite outras operações

    // saveFrame() ;
    // myMovie.save("bbb.tif") ;
    myImage.save("ccc.tif") ;
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Áudio

Usando a biblioteca Minim

<http://code.compartmental.net/tools/minim/manual-audioeffect/>

```
import ddf.minim.*;
AudioPlayer player;
Minim minim;

void setup()
{
    size(512, 200, P2D);
    minim = new Minim(this);
    player = minim.loadFile("groove.mp3");
    player.play();
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Áudio

Filtros áudio (usando a classe AudioEffect da Minim):  
inverte a sequência de amostras áudio e devolve o buffer alterado

```
class ReverseEffect implements AudioEffect {
    void process(float[] samp)
    {
        float[] reversed = new float[samp.length];
        int i = samp.length - 1;
        for (int j = 0; j < reversed.length; i--, j++)
            reversed[j] = samp[i];
        arraycopy(reversed, samp);
    }

    void process(float[] left, float[] right) {
        process(left);
        process(right);
    }
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Áudio

Aplicação do filtro (método addEffect)

```
import ddf.minim.*;
import ddf.minim.effects.*;

Minim minim;
AudioPlayer player;
ReverseEffect reffect;

void setup() {
    size(512, 200, P2D);
    minim = new Minim(this);
    player = minim.loadFile("groove.mp3", 2048);
    player.loop();
    reffect = new ReverseEffect();
    player.addEffect(reffect);
}
```

PCM (DI/FCT/UNL) ©nmc 2012

# XML

XML (eXtensible Markup Language)

- W3C (WWW Consortium)
- Meta-linguagem para definição de linguagens de *markup* (e.g. *HTML*, *SVG*)
- Conformidade e interoperabilidade com SGML, mais simples
- Utilizado em metadata, por exemplo MPEG-7, para descrever informação vídeo.

PCM (DI/FCT/UNL) ©nmc 2012

# XML (Exemplo)

```
<media_library>
  <item type = "image">
    <source> image1.jpg </source>
  </item>
  <item type = "image">
    <source> image2.jpg </source>
  </item>
</media_library>
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – XML

Classe XMLElement (Processing 1.5): Representação de um objecto XML, inclui métodos para acesso à lista de filhos

```
XMLElement xml;

void setup() {
  size(200, 200);
  xml = new XMLElement(this, "riddles.xml");
  int numRiddles = xml.getChildCount();
  for (int i = 0; i < numRiddles; i++) {
    XMLElement kid = xml.getChild(i);
    int id = kid.getInt("id");
    String question = kid.getString("question");
    String answer = kid.getContent();
    println(id + " : " + question + " : " + answer);
  }
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – XML

Exemplo para código anterior

```
<?xml version="1.0"?>
<riddles>
  <riddle id="0" question="Who is the director of ET?">Steven
    Spielberg</riddle>
  <riddle id="1" question="When was Blade Runner released?">1982</riddle>
</riddles>
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Captura de Vídeo

Captura de vídeo: class Capture para captura e manipulação de vídeo a partir da uma câmara

```
import processing.video.*;
Capture cam;

void setup() {
  size(640, 480);
  // If no device is specified, will just use the default.
  cam = new Capture(this, 320, 240);
}

void draw() {
  if (cam.available() == true) {
    cam.read();
    image(cam, 160, 100);
  }
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Processing – Captura de Vídeo

Diferença de imagens: exemplo de aplicação de captura

```
void draw() {
  for (int i = 0; i < numPixels; i++) {
    color currColor = video.pixels[i];
    color prevColor = previousFrame[i];
    int currR = (currColor >> 16) & 0xFF; // Like red(), but faster
    int currG = (currColor >> 8) & 0xFF;
    int currB = currColor & 0xFF;
    //...
    int diffR = abs(currR - prevR);
    pixels[i] = color(diffR, diffG, diffB); // image with the
    // differences
    //...
  }
}
```

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

- Cada aplicação multimédia tem uma estrutura e objectivos, requerendo características que dependem da mensagem a ser transmitida.
- Propriedades para avaliação e selecção de uma authoring tool:
  1. Modelo e paradigma para organização do conteúdo
  2. Ferramentas de edição de conteúdos
  3. Programação
  4. Interactividade
  5. Performance
  6. Reprodução da aplicação
  7. Entrega da aplicação
  8. Plataformas

*PCM (DI/FCT/UNL) ©nmc 2012*

## Sistemas de Autoria (Avaliação)

### 1. Modelo e paradigma para organização do conteúdo

Uma vez que a maior parte dos paradigmas permite incluir as propriedades de outros, a escolha depende de:

- Paradigma mais natural para a aplicação
- Complexidade da aplicação
- Preferências e experiência do autor

*PCM (DI/FCT/UNL) ©nmc 2012*

## Sistemas de Autoria (Avaliação)

### 2. Ferramentas de edição de conteúdos

- A ferramenta inclui funcionalidade de conteúdo? Para ajustar conteúdo quando se integram vários media (por exemplo, imagens, texto, gráficos e animações)
- Habitualmente um sub-conjunto das funcionalidades existentes em ferramentas especializadas.

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

### 3. Programação

#### Escolha de ferramentas:

- Ferramenta de autoria: aproximação simplificada
- Programação multimédia: mais complexo

#### Aproximações de programação:

- Programação visual : simples, mais lento
- Linguagens de script (ActionScript): mais poderoso e versátil
- Linguagens de prototipagem (Processing): protótipos mas bastante sofisticados
- Linguagens de programação tradicionais: ainda mais complexo + Arquitecturas de Software para Multimédia  
Por exemplo: MS Direct X, openFrameworks

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

### 3. Programação

- Os scripts permitem a manipulação, personalização e automação de funcionalidades pré-definidas
- Mais comandos = ferramentas mais poderosas
- Mais fácil aprender uma linguagem de script depois de outra
- Características apreciadas: *debugging*, edição de texto, ajuda online, inclusão de texto pré-formatado e estruturado, pesquisa, extensões (por exemplo DLLs).

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

### 4. Interactividade

Permite flexibilidade e controlo da apresentação do conteúdo e do fluxo de informação.

São suportados diferentes níveis de interactividade:

- Acesso simples: acesso a uma parte do conteúdo
- Acesso condicional: depende de uma condição ou evento
- Decisão estruturada: condições e gestão de eventos mais complexos

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

### 5. Desempenho

Aplicações mais complexas requerem sincronização precisa de eventos (por exemplo corrida de carros realista). A ferramenta de autoria suporta?

- Permite a especificação de diferentes velocidades para diferentes processadores ou capacidade de memória.
- Usa linguagens de script ou de programação para assegurar a temporização adequada dos eventos em cada situação.

### 6. Reprodução da aplicação

- Suporta reprodução para testar o *layout*, o desempenho e a usabilidade durante o desenvolvimento.

PCM (DI/FCT/UNL) ©nmc 2012

## Sistemas de Autoria (Avaliação)

### 7. Entrega da aplicação

- Permite a produção de um ambiente de suporte de execução que não necessite da ferramenta de autoria
- Não necessita da instalação da ferramenta de autoria
- Não permite o acesso ao código fonte, nem a alteração do conteúdo da aplicação.

### 8. Plataformas

Escolher uma ferramenta de autoria que:

- Permita a transição simples entre plataformas
- É compatível ou fornece ambientes de execução para as plataformas mais comuns.

PCM (DI/FCT/UNL) ©nmc 2012

## PCM

# 3

## Design Multimédia

PCM (DI/FCT/UNL) ©nmc 2012

## Design Multimédia

“A **rule** says ‘you must do it this way.’  
A **principle** says ‘this works  
... and has through all remembered time’  
Anxious, **inexperienced** writers obey rules.  
**Rebellious**, unschooled writers break rules.  
**Artists** master the form”

Robert McKee

PCM (DI/FCT/UNL) ©nmc 2012



## Design Multimédia

**Função . Usabilidade . Estética**

Don Norman

*PCM (DI/FCT/UNL) ©nmc 2012*

## Design Multimédia

- Design envolve concepção da funcionalidade e do aspecto da aplicação multimédia
  - INPUT: Requisitos e limitações das tecnologias
  - OUTPUT: Storyboard/script e protótipos das interfaces
    - Estrutura da aplicação
    - Cada ecrã ou unidade
    - Interface utilizador
- Protótipos permitem designs alternativos e avaliação com utilizadores

PCM (DI/FCT/UNL) ©nmc 2012

## Design Multimédia

- Estrutura
  - Estruturas hipermédia ou sequências ecrã/cenário
- Conteúdo
  - Elementos multimédia e *layout* no tempo e no espaço + características técnicas
- Interação e navegação

PCM (DI/FCT/UNL) ©nmc 2012

## Design da Estrutura

- Estrutura de "ecrãs"
- Condiciona a mensagem e a experiência interactiva
- Design da navegação define os percursos do utilizador
- Paradigmas:
  - Temas segundo um tópico
  - Cronológica
  - Métodos híbridos
- Mapa de navegação
  - Fornece um índice gráfico do fluxo lógico
  - Descreve as hiperligações
  - Ilustra as interacções

PCM (DI/FCT/UNL) ©nmc 2012

## Estruturas de Navegação

- **Linear:** Sequência de ecrãs.
- **Hierárquica:** Navegação ao longo dos ramos de uma árvore.
- **Não linear:** O utilizador navega livremente, podendo escolher o percurso em cada instante.
- **Composta:** O utilizador pode navegar livremente mas tem zonas de restrição (sequências lineares por exemplo).

PCM (DI/FCT/UNL) ©nmc 2012

## Design do Conteúdo

- Esboços de cada ecrã:
  - Descrições textuais e esquemas
  - Especificando cada imagem, bloco de texto, objecto gráfico, incluindo dimensões e durações.
  - Detalhando os elementos de navegação
  - Especificando os detalhes, incluindo cores, sombras, formatos...

PCM (DI/FCT/UNL) ©nmc 2012

## Design do Conteúdo

- Layout: Ecrãs simples e leves, com margens largas para assinalar aspectos importantes.
- Usualmente apenas dois níveis, de modo a chamar a atenção
- Tamanhos, normalmente limitados a quatro. Tamanhos maiores para atrair a atenção.
- *Fonts*, normalmente limitadas a três.
- Outros efeitos...com cuidado e em áreas limitadas.

PCM (DI/FCT/UNL) ©nmc 2012

## Utilização da Cor

Recomendações (Dix, Faulkner):

- As cores utilizadas nas interfaces devem ser tão distintas quanto possível
- Não serem afectadas pelo contraste
- A cor azul não deve ser utilizada para apresentar informação crítica
- A cor deve ser complementada com outros indicadores, para contemplar utilizadores com deficiências.
- Devem-se respeitar as convenções aceites e as expectativas do utilizador. O vermelho, verde e o amarelo são usados para parar, avançar e esperar. Vermelho para alarme e emergência, verde normal e amarelo para espera.

*PCM (DI/FCT/UNL) ©nmc 2012*

## Utilização da Cor

- As cores têm significado por associação cultural...

Pureza, inocência, limpeza



Calor, violência, guerra, parar, proibido



Natureza, saúde, inveja, avançar, permitido



Frio, profundidade, sabedoria, confiança



Calor, alegria/ciúme, doença



Realeza, mistério, romance, nostalgia

*PCM (DI/FCT/UNL) ©nmc 2012*

## Erros mais Comuns

- Cores berrantes e mistura de muitas cores.
- Ecrãs repletos de informação textual e gráfica.
- Humor vulgar ou animações repetitivas.
- Sons retumbantes e repetitivos.
- Molduras grossas e com padrões.
- Citações famosas ou de filmes e/ou best-sellers.
- Estruturas de navegação com mais de dois cliques para sair da aplicação.

PCM (DI/FCT/UNL) ©nmc 2012

## Design da Interface Utilizador

- Combina os conteúdos multimédia e o sistema de navegação
- Interface deve aumentar o desempenho...
  - Se for esse o propósito da aplicação
- Usar metáforas conhecidas dos utilizadores
- Linhas de comando, menus, linguagem natural, diálogos, formulários, WIMP, 3D, gestos...
- Interface deve ser acessível
- Ergonomia inclui factores humanos e características físicas

PCM (DI/FCT/UNL) ©nmc 2012

## Design de Elementos Interactivos

Funções essenciais:

- Sair da aplicação em qualquer altura
- Cancelar uma actividade ou anular uma escolha
- Aceder a um mapa de navegação global da aplicação multimédia interactiva

*PCM (DI/FCT/UNL) ©nmc 2012*

## Organização de Controlos

- Funcional: Organizados por forma a aproximar os que se relacionam em termos funcionais.
- Sequencial: Organizados de forma a reflectir a ordem de uma interacção típica.
- Frequência: Organizados de acordo com a frequência de utilização.

*PCM (DI/FCT/UNL) ©nmc 2012*

## Navegação e Interação

Interfaces para navegação

- Links embebidos
- Barras de navegação
- *Breadcrumbs*  
(por exemplo, home>multimedia>design>color)
- Links de navegação em sequência  
(por exemplo prev 1 2 3 next)
- Mapas de navegação, História, Caminhos, *Bookmarks*, Pesquisa...

PCM (DI/FCT/UNL) ©nmc 2012

## Design Multimédia

Orientações...

- 10 Nielsen's Usability Heuristics (Interfaces)
- Schneiderman's Golden Rules (Interfaces)
- Web Design Guidelines (Web)

PCM (DI/FCT/UNL) ©nmc 2012

## Regras de Usabilidade

Nielsen Version 2.0, 1994

1. **Visibility of system status**
2. Match between system and the real world
3. User control and freedom
4. **Consistency and standards**
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. *Aesthetic* and minimalist design
9. Help users recognize, diagnose & recover from errors
10. Help and documentation

PCM (DI/FCT/UNL) ©nmc 2012

## Regras de Usabilidade

Schneiderman

1. **Strive for consistency**
2. Enable frequent users to use shortcuts
3. **Offer informative feedback**
4. Design dialogues to yield closure
5. Offer simple error handling
6. **Permit easy reversal of actions**
7. Support internal locus of control
8. Reduce short-term memory load

PCM (DI/FCT/UNL) ©nmc 2012

## Regras de Usabilidade

### Web Design

1. Put the user first (also accessibility)
- 2. Put the user in control**
3. Don't provide too much choice
- 4. Don't make assumptions about users' behavior**
5. Use technology judiciously
- 6. Understand your site's context**
7. Keep up with change
8. Don't neglect aesthetics
9. Know your limitations

PCM (DI/FCT/UNL) ©nmc 2012

## Integração Multimédia

- O conflito entre tipos de informação, por exemplo texto e imagem, é comum na história da escrita.
- A tecnologia e a representação digital permite a integração mas coloca novos desafios...
- ...em particular a relação entre informação apresentada de forma **espacial** (texto, imagens) e **temporal** (animação, áudio, vídeo)

PCM (DI/FCT/UNL) ©nmc 2012

## Informação Estática e Dinâmica

- Media Dinâmicos:
  - Sequência temporal
- Multimédia:
  - Sequência temporal, sincronização, integração com informação estática
- Hipermedia
  - Estrutura temporal
  - Aspectos retóricos e estéticos
  - Novos problemas na navegação

PCM (DI/FCT/UNL) ©nmc 2012

## Informação Estática e Dinâmica

- Requerem atitudes diferentes
- Informação estática
  - Linearidade subjectiva
  - Mais activo (por exemplo, leitura)
- Informação dinâmica
  - Linearidade objectiva
  - Mais passivo (por exemplo, ver filme)
- Como integrar?

PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade

- Liestøl [ECHT94] sugere continuidade tal como na técnica de edição de continuidade no cinema
- Evitar descontinuidades ao passar de um meio a outro
  - Exemplo: Forma passiva/experiencial de ver vídeo para a leitura activa/reflexiva de um livro
- Resulta num conjunto de regras:
  - Dimensão topográfica para o vídeo
  - Atribuição de propriedades temporais ao texto
  - Narração para efeito de continuidade

*PCM (DI/FCT/UNL) ©nmc 2012*

## Continuidade

Inicialmente:

- Modo dominante influenciado pelo teatro
- Organizado por cenas (longas)
- Filmado sempre da mesma forma, com câmara estável
- Insuficiente e pouco interessante depois de algum tempo

*PCM (DI/FCT/UNL) ©nmc 2012*

## Continuidade e Edição

Cinema como arte (D.W. Griffith):

- Alteração do ângulo de filmagem.
- Utilização de *close ups* com efeitos dramáticos.
- Utilização do **shot** como elemento básico.
- A edição passou a ser a técnica de produção de filmes.



PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade e Edição

Lev Kuleshov, montagem:

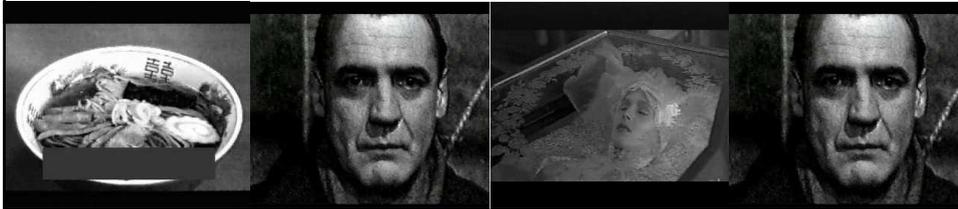
- Ligação entre *shots* de planos e situações que ocorrem em espaços e tempos diferentes.
- Construção de **significado**.
- Influenciou outros nomeadamente Eisenstein.



PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade e Edição

- Efeito Kuleshov: poder manipulativo da montagem, independentemente do conteúdo.
- Actor vs. sopa = fome // caixão = tristeza...
- A falta de filme obrigou à reutilização...



PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade e Edição

- Edição
  - É disruptiva quando provoca quebras no tempo e no espaço
- Descontinuidade
  - Quando há shots de tamanhos diferentes, conteúdo e aparência visual diferente
- Pode perturbar os espectadores (utilizadores)...

PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade e Edição

Edição de continuidade permite experiência sem cortes (visíveis), com narrativa coerente:

- Espacial: Direcção, espaço
- Gráfica: Objectos, luz, tom, acções
- Temporal: Cronologia, ordem, frequência, duração
- Rítmica: Acção, complexidade de informação

*PCM (DI/FCT/UNL) ©nmc 2012*

## Continuidade em Hipermedia

- Ligação entre nós (hipermedia) audiovisuais, é semelhante a edição de vídeo (shots ou cenas como nós).
  - Mas passa a ser o utilizador a controlar...
- Problema novo:
  - Ligação entre informação dinâmica e estática

*PCM (DI/FCT/UNL) ©nmc 2012*

## Continuidade em Hipermedia

- Ligação entre nós (hipermedia) audiovisuais, é semelhante a edição de vídeo (shots ou cenas como nós).
  - Mas passa a ser o utilizador a controlar...
- Problema novo:
  - Ligação entre informação dinâmica e estática

PCM (DI/FCT/UNL) ©nmc 2012

## Continuidade em Hipermedia

- Equilíbrio difícil entre media e acção do utilizador
  - Especialmente na passagem de dinâmico a estático
- Seguir hiperligações envolve participação activa
- Descontinuidade pode interferir com o fluxo das narrativas
- Pode adaptar-se edição de continuidade [Liestøl, Hypercafé]
  - Video como footnotes, texto narrado...

PCM (DI/FCT/UNL) ©nmc 2012

## Novos Problemas Design Multimédia

- Potencial para informação excessiva e desorientação
- Orientações de design e suporte tecnológico
- Informação sobre hiperligações (*link awareness*) (destino/origem/duração)
- Como fazer âncoras? Mudança de cor, animação, texto...

PCM (DI/FCT/UNL) ©nmc 2012

## Exemplo - Hipervídeo

Convergência de filme e hipermédia

- Não linear como hipermédia
- Imagens em movimento tal como no filme
- Significados através da montagem
- Gera múltiplas ficções
- Altera a narrativa do filme e também a forma como se visualiza
- O utilizador cria alternativas através de escolhas e interacção.
- Dá mais possibilidades ao utilizador.

PCM (DI/FCT/UNL) ©nmc 2012