



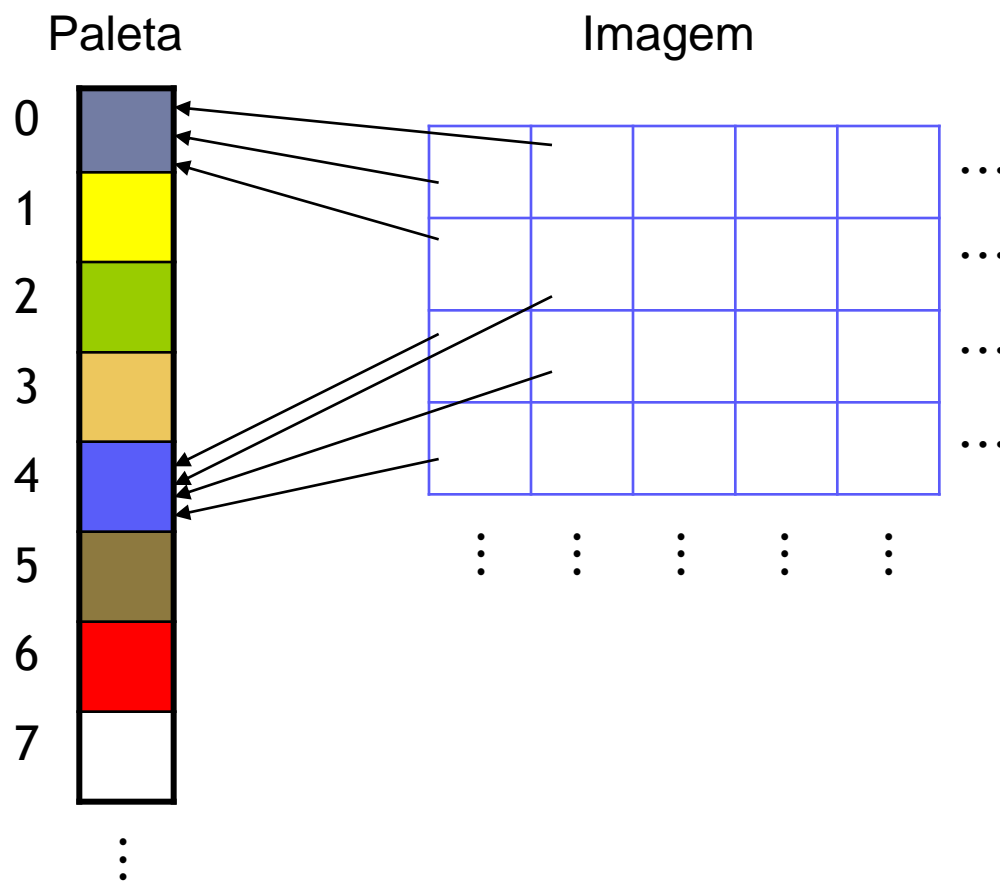
- Paleta de Cores (*Color index*)
- *Tiles*
- *Tilesets*
- *Tilemaps*
- *Sprites*

# Paleta de cores

3

- **Do ponto de vista artístico**
  - ▣ Criar uma família de cores harmônicas
  - ▣ Existem diferentes tipos:
    - *Cartoon, tons pastéis, noturna,...*
- **Do ponto de vista computacional**
  - ▣ Economizar memória
  - ▣ Efetuar efeitos de cores

# Imagens tipo indexed-color



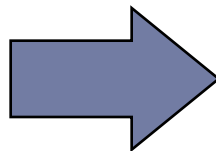
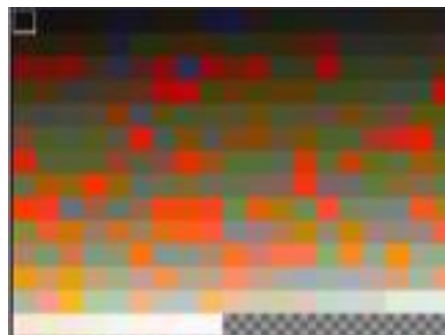
Números da economia:

- Imagem de 200 x 200
- 24 bits:  $200 \times 200 \times 3 \approx 117 \text{ Kb}$
- Paleta 8 bits:
- $768 + 200 \times 200 \approx 40 \text{ Kb}$

# Programando com imagens

5

- Imagens tipo indexed-color



# Paleta de cores

6



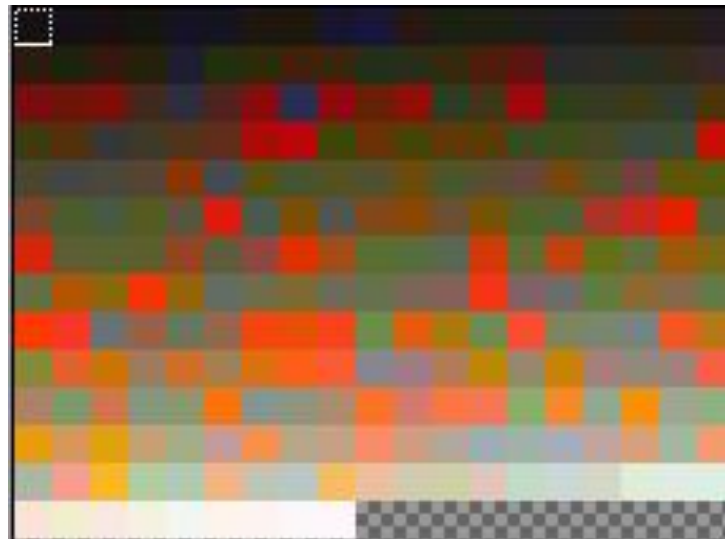
- Dimensão: 240x360
- 86400 pixels
- Cada pixel usa três canais (RGB).
- Cada canal requer 1 byte
- Resulta em 259200 bytes
- 253,125Kb

# Paleta de cores

7














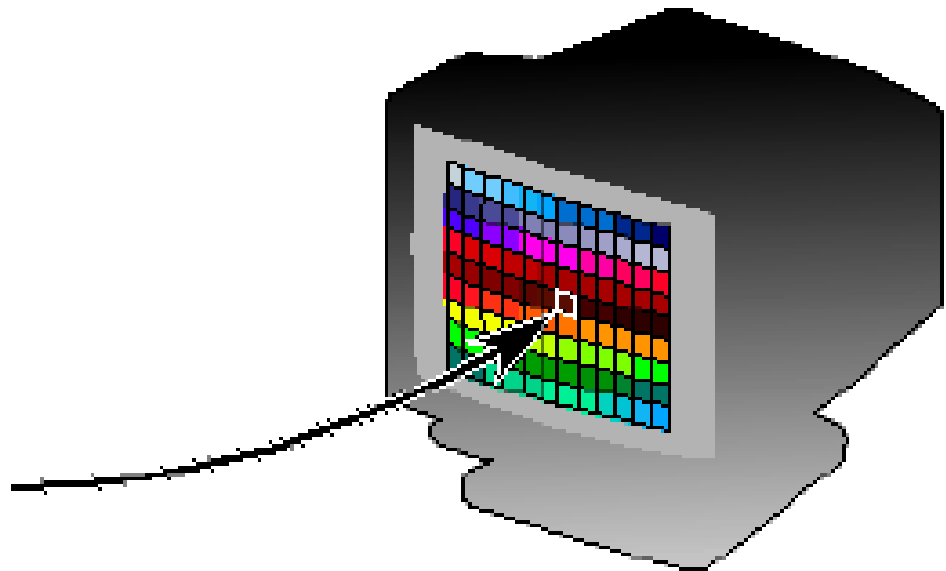
- Dimensão: 240x360
- 86400 pixels
- Usa uma paleta com 256 cores



# Paleta de cores

8

Index	Red	Green	Blue
0			
1			
2			
3			
4			
5			
...			
...			
...			
296			
...			
...			
...			





# Paleta de cores

9



- Dimensão: 240x360
- 86400 pixels
- Usa uma paleta com 256 cores
- Cada entrada da paleta tem 3 bytes (RGB)
  - ▣ 768 bytes
- Para cada pixel da imagem é usado 1 byte (índice da paleta)
  - ▣ 84,375 Kb
- Total: 85,125Kb (**~3x menor**)

# Paleta de cores

10

- Como construir?
  - Como transformar as  $2^{24}$  cores em apenas 256?
  - Percorrer a imagem, e para cada pixel diferente, adicionar uma entrada na paleta e indexá-lo

# Paleta de cores

11

- As paletas podem ser pré-definidas pelos sistemas. O que retira a necessidade de armazenamento da mesma.
  - ▣ Como o padrão “IBM VGA palette”:  
[http://www.fileformat.info/mirror/egff/ch02\\_02.htm](http://www.fileformat.info/mirror/egff/ch02_02.htm)
- Mas as melhores paletas são aquelas otimizadas a partir imagem que elas representam. As  $n$  cores que melhor representam a imagem. Chamada de paleta otimizada.
- Das 16 milhões de cores a paleta utiliza apenas **256** cores, obviamente, que há uma perda de qualidade de informação, em favorecimento a economia de armazenamento.

# Paleta de cores

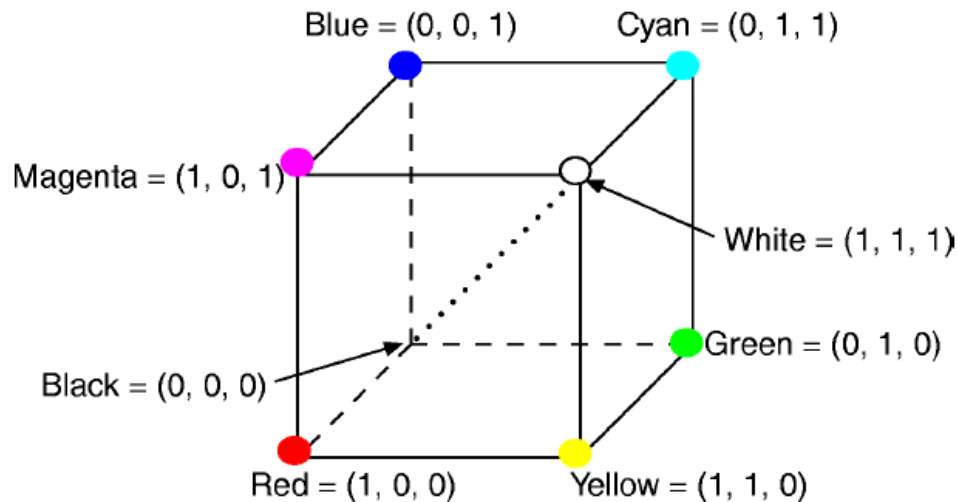
12

- É necessário fazermos um processo de quantização de cores para definir as 256 melhores cores que compõem a nossa paleta de cores de uma determinada imagem.
- Um histograma pode nos dar as cores mais freqüentes, e podemos aproximar as outras pelo cálculo da **distância de cores**
- Muitas vezes a redução não afeta a qualidade da imagem ou o resultado é aceitável.

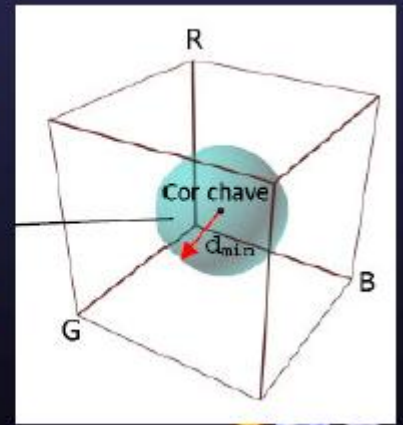
# Paleta de cores

13

## □ Distância de cores RGB



Cor-chave é um raio que delimita uma região de proximidade. Não é possível garantir que, em todas as direções, as cores nos limites da esfera sejam similares a cor-chave.



# Paleta de cores

14

- Distância de cores RGB

- Cálculo da *distância euclidiana*

$$d = \sqrt{(c_R - o_R)^2 + (c_G - o_G)^2 + (c_B - o_B)^2}$$

- $c$  é a cor chave (que queremos comparar)
    - $o$  é a cor que está sendo testada
    - R, G e B são as componentes
    - $d$  é a distância: entre  $c$  e  $o$ . Quanto maior  $d$ , mais “diferentes” as cores são

# Paleta de cores

15

## □ Distância de cores RGB

### ■ Proporção:

- $d_{max}$  é a distância entre o valor mínimo e máximo (0 e 255)

$$d_{max} = \sqrt{(0-255)^2 + (0-255)^2 + (0-255)^2}$$

$$d_{max} = \sqrt{195075}$$

$$d_{max} \approx 441,673$$

- Fazemos uma proporção entre a distância e  $d_{max}$ , e checamos se está dentro de nosso raio de tolerância

$$\Delta d = \frac{d}{d_{max}}$$

# Paleta de cores

16

## □ Distância de cores RGB

*No código-fonte...:*

```
dd = d / dmax;  
if(dd <= tolerancia){  
    // faz alguma coisa  
}
```

- ▣ Neste caso,  $dd$  é um número entre 0 e 1



# Tiles

17

- Uma pequena porção de imagem
  - ▣ Metáfora: Azulejo, tijolo ou um carimbo
  - ▣ A cena é formada por uma coleção de *tiles*
  - ▣ Cada *pixel do tile utiliza um índice da paleta*



# Tiles e Paleta de Cores

18

0	1	1	2	3	0	0	0	1	1	2
0	0	1	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	1	1
3	3	1	2	1	1	1	1	1	1	1
0	0	3	1	1	1	1	1	1	0	0



- Cada número é um índice da paleta!

# Tileset

19

- Coleção de *tiles*
- Em geral organizados em uma matriz ou um vetor
- Cada *tile* em um *tileset* possui um identificador (ID)
  - ▣ Um código numérico



# Tileset

20



# Tilemap

21

- Uma imagem criada através da composição de diversos ***tiles de um tileset***.
- Objetivo: economizar memória!



# Tilemap

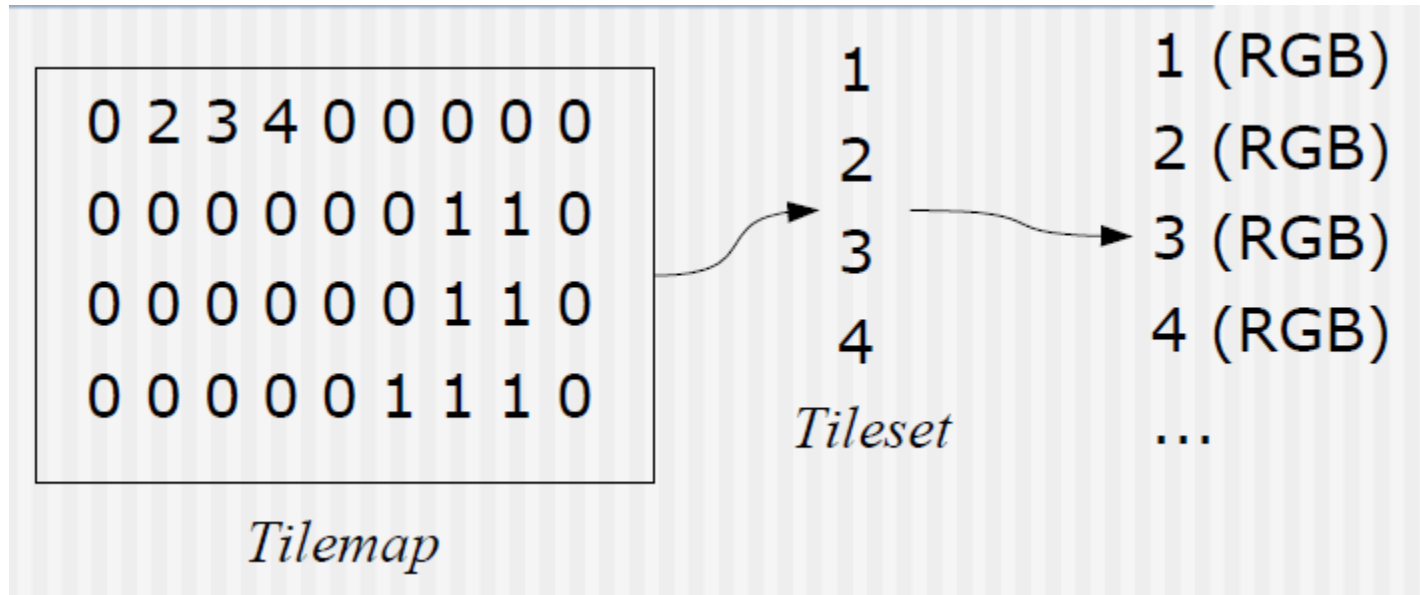
22

0	2	3	4	0	0	0	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	1	0

- Cada número é um índice do *tile no tileset*

# Tilemap, Tiles e Paleta de cores

23

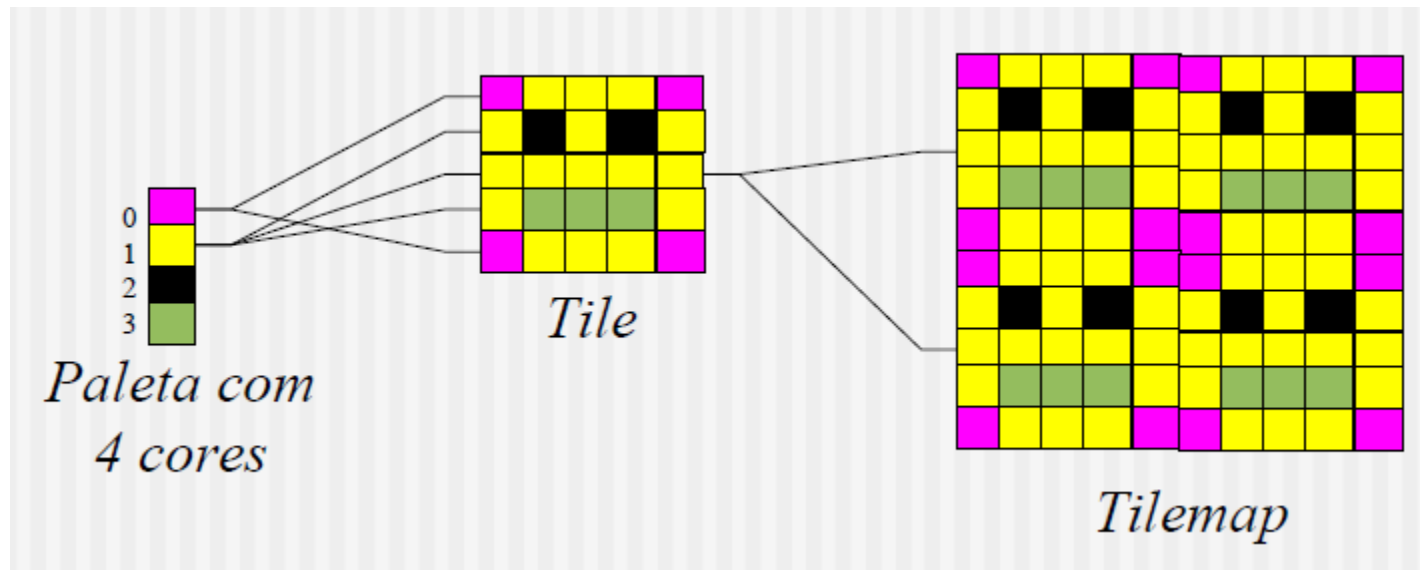


- Cada número no *tilemap* é um índice do tile no *tileset*
- Cada *pixel* do tile no *tileset* é um índice da paleta



# Tilemap, Tiles e Paleta de cores

24



- Cada número no *tilemap* é um índice do tile no *tileset*
- Cada *pixel* do tile no *tileset* é um índice da paleta



- **Cenário 1: Sem tilemap**
  - ▣ Imagem com 2000x1000
  - ▣ Sistema de cor RGB (cada pixel com 24 bits)
  - ▣ Imagem com **5,72 Mb**
- **Cenário 2: Com *tilemap*, *tileset* e *tiles***
  - ▣ Imagem com 2000x1000
  - ▣ Uma paleta com 256 cores (0,75 Kb)
  - ▣ Cada tile com as dimensões de 20x20 pixels
    - Matriz 100x50 de bytes (ID do tile) (4,88Kb)
  - ▣ Tileset com 50 tiles (19,53Kb - 20x20x50)
  - ▣ **Total: 25,16Kb (menor ~233 vezes)**

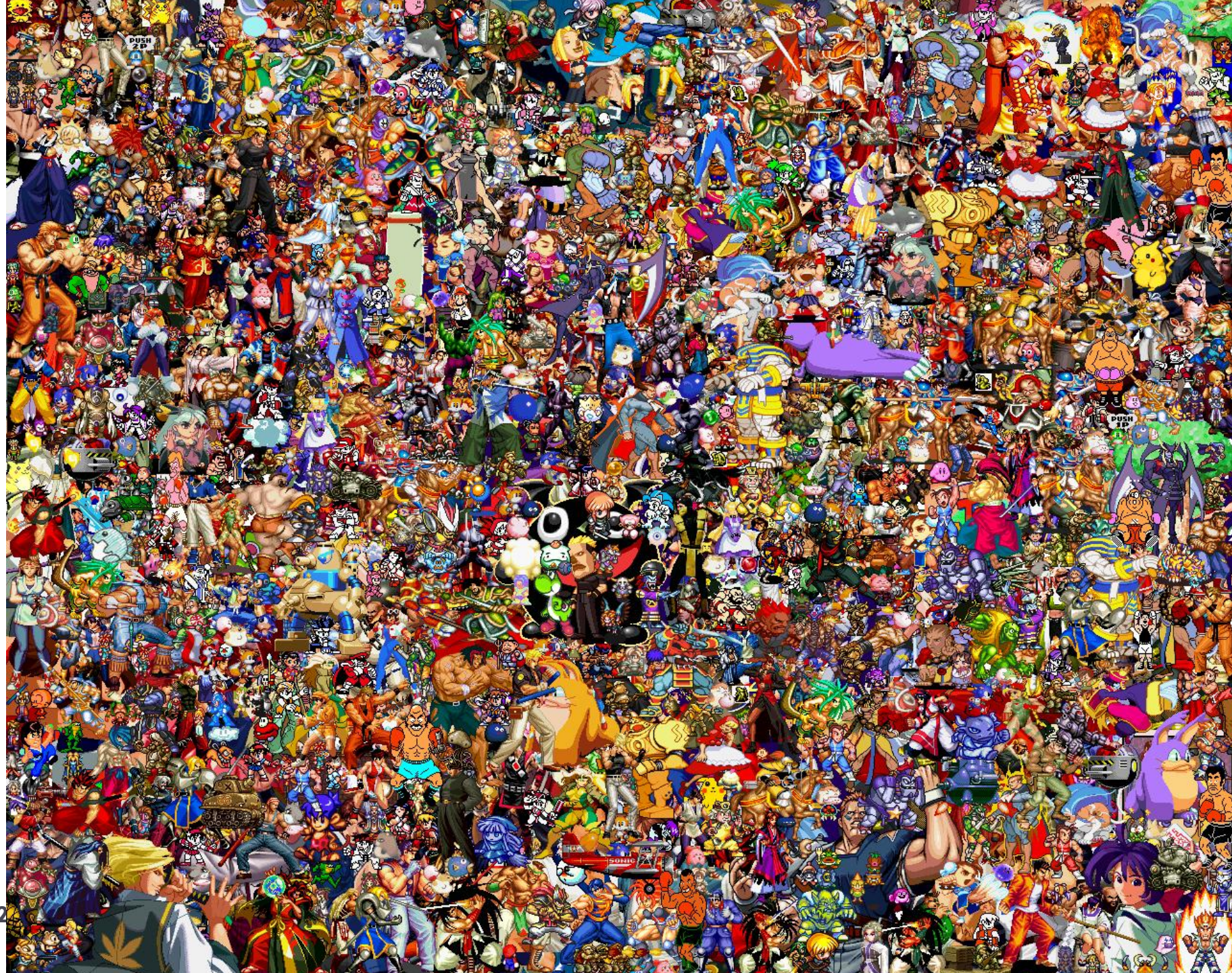
# Sprites

26

- “Fantasma”, espectro, ...
- Animação clássica – sequencia de quadros que serão exibidos conforme a ação do jogador ou mudança de estado de algum objeto
- Cada vez que atualiza a tela pinta um novo *frame*
- Usado para:  
Personagem principal, inimigos, itens e artefatos no cenário





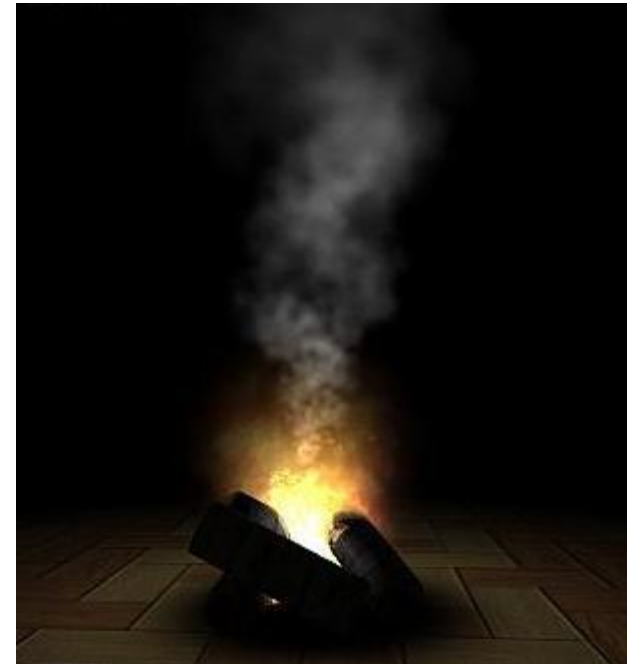




# Sprites

28

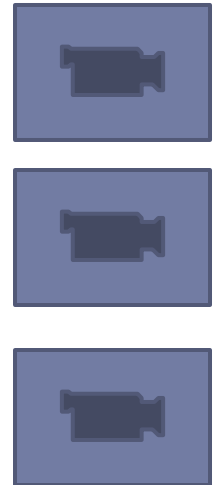
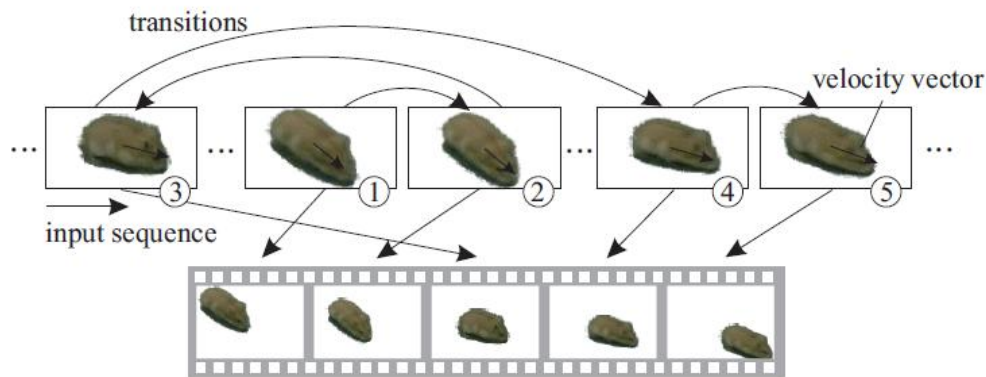
- Fenômenos *fuzzy*
  - ▣ Nuvens, fogo, fumaça...
  - ▣ Associado a simulação de partículas



# Técnicas para a animação de *sprites*

29

- Animações criadas por uma reorganização dos frames previamente gravados



Schödl, A. and Essa, I. A. 2002. Controlled animation of video sprites. In *Proceedings of the 2002 ACM Siggraph/Eurographics Symposium on Computer Animation* (San Antonio, Texas, July 21 - 22, 2002). SCA '02. ACM, New York, NY, 121-127

15,000 usable sprite frames!

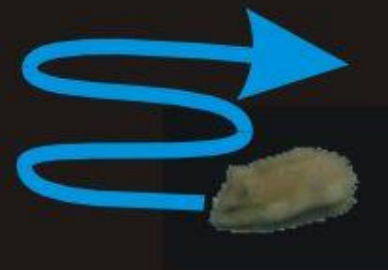
# What are video sprites?



Data capture



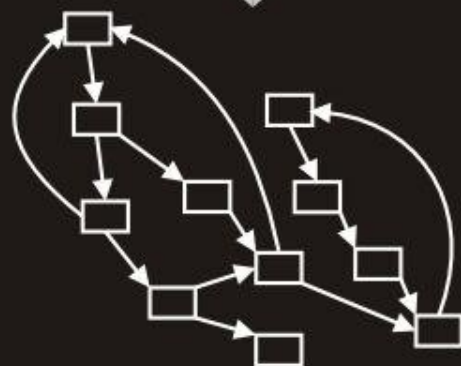
Train similarity classifier



Constraints,  
e. g. motion trajectory



Extract sprites  
using chromakeying



Find transitions  
by comparing  
all pairs of frames



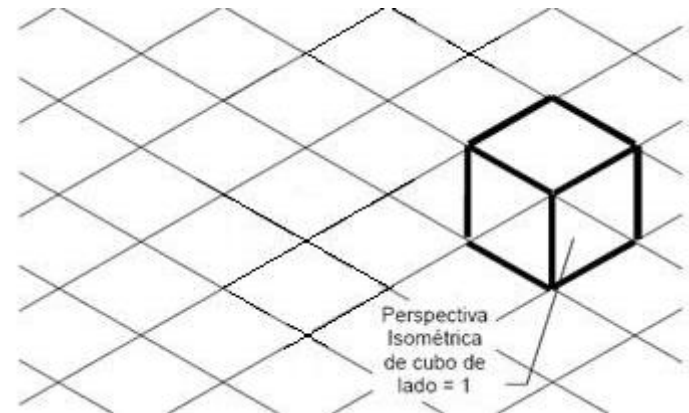
Find sequence of frames  $s_1...s_n$   
that shows desired animation

Render and composite

# Visão Isométrica

31

- Isométrico: método matemático para desenhar uma figura 3D sem o uso da perspectiva.
- Todos os tamanhos são preservados e os objetos são desenhados com uma inclinação de  $30^\circ$ .
- Os sprites também devem ser mudados.
- Também conhecido como “2.5D” ou “falso 3D”



# Isometria

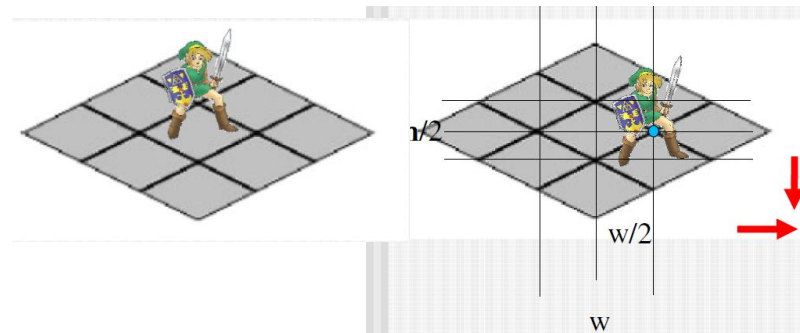
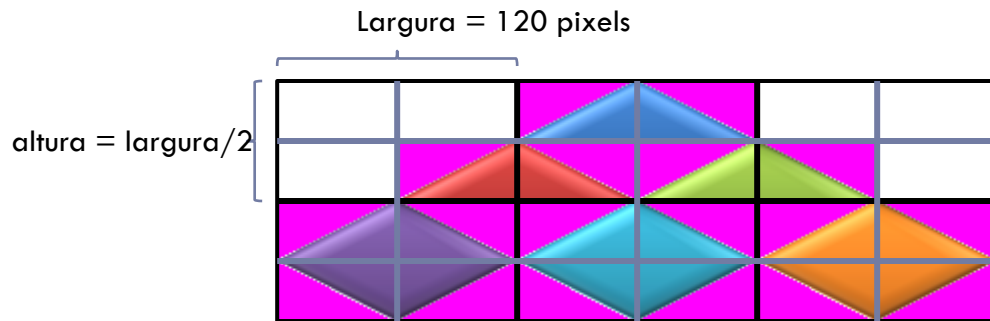
- Caso especial em que o plano de projeção forma o mesmo ângulo com os três eixos principais. As projeções dos três vetores unitários canônicos formam ângulos de  $120^\circ$  entre si.
  - Isto permite que as medições feitas na projeção em cada eixo utilize a mesma escala
-



# Visão Isométrica

33

- Tiles isométricos
  - ▣ Como resolver a sobreposição?
  - ▣ Como navegar??



# Referências

34

- Materiais do professor Leandro Tonietto e João Bittencourt