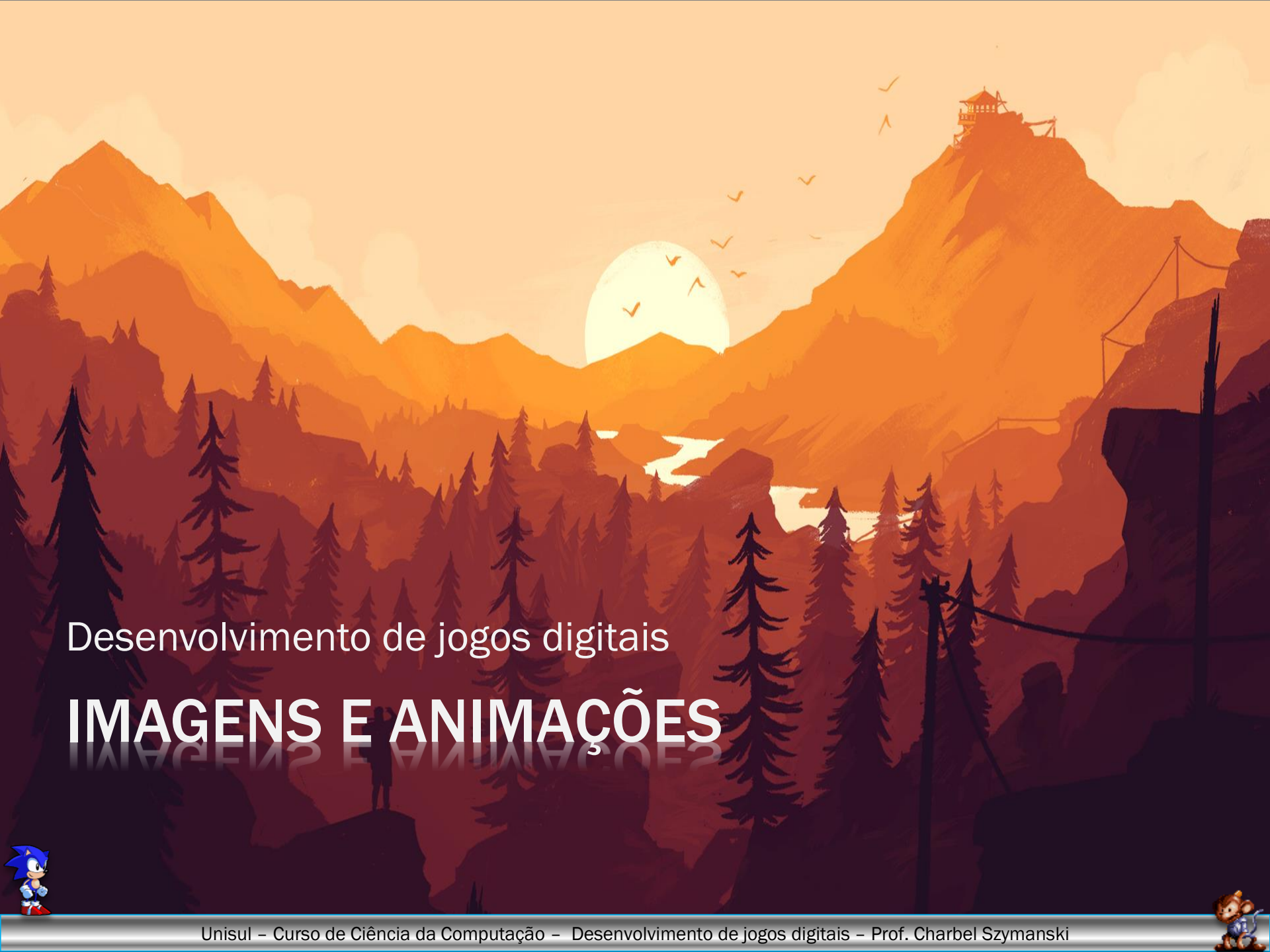


Desenvolvimento de jogos digitais

IMAGENS E ANIMAÇÕES

SUMÁRIO

- Tipos de imagem
- Manipulação de imagens
- Temporização
- Animações 2D

SUMÁRIO

- **Tipos de imagem**
- Manipulação de imagens
- Temporização
- Animações

TIPOS DE IMAGEM

- Sem compactação
- Compactação sem perda
- Compactação com perda

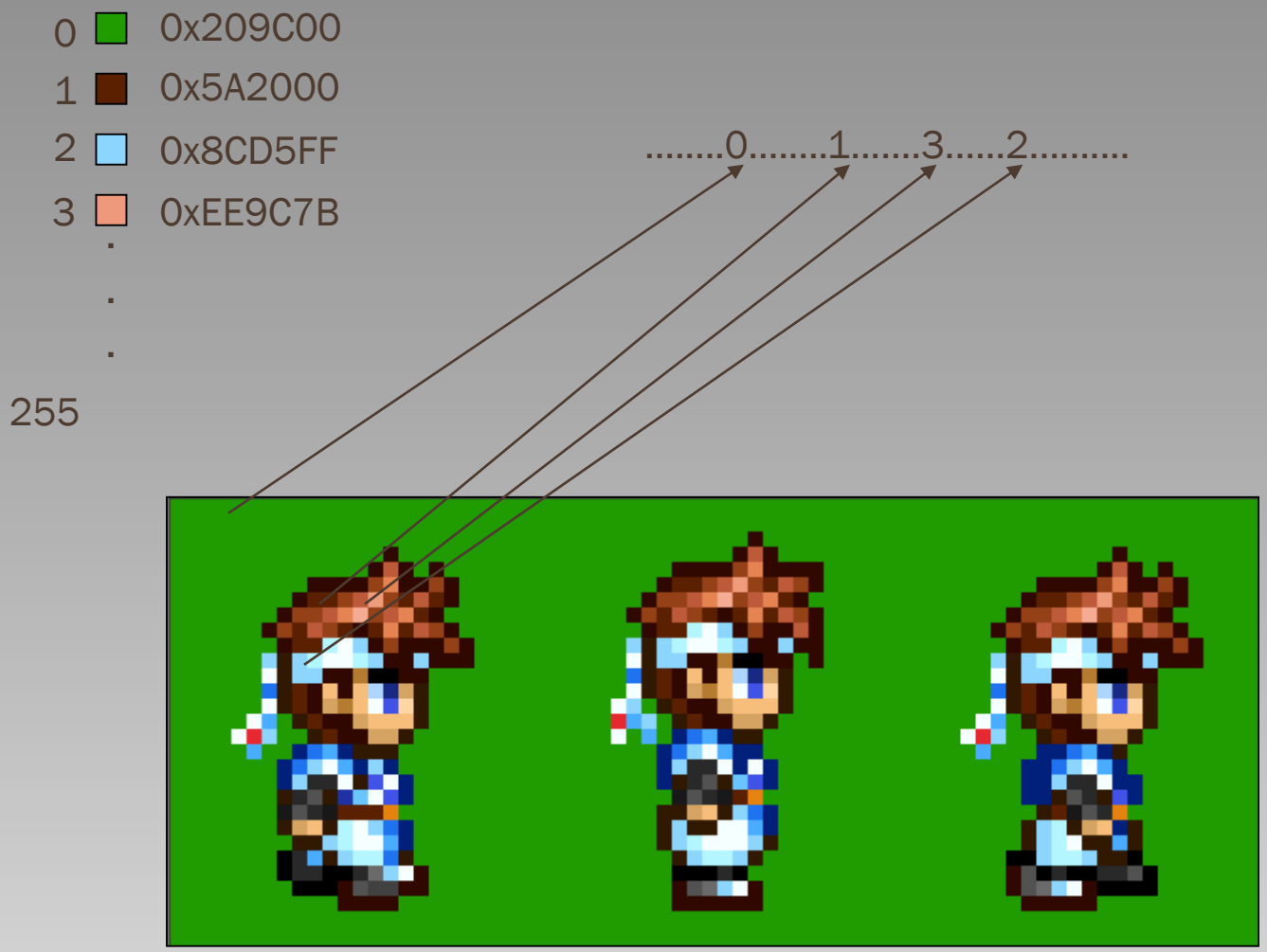
TIPOS DE IMAGEM

- Um dos formatos sem compactação mais utilizados é Bitmap (BMP)
- Os formatos de compactação sem perda mais usados são GIF e PNG
- O formato de compactação com perda mais usado é JPG

IMAGENS PALETIZADAS

- Imagens paletizadas são imagens onde a informação de cores (RGB) dos pixels não é gravada diretamente para cada pixel. Nestas imagens grava-se no início do arquivo a paleta em RGB, essa paleta pode ser de 16 ou 256 cores, em seguida grava-se a informação de cor para cada pixel. **Essa informação de cor é somente um índice para a paleta.**

IMAGENS PALETIZADAS



IMAGENS PALETIZADAS

- Em imagens paletizadas com paleta de 256 cores cada pixel ocupa um Byte, pois deve guardar um valor de 0 a 255, que representa o índice da cor utilizada pelo pixel.
- Numa imagem paletizada com paleta de 16 cores cada pixel da imagem é uma informação de 4 bits pois a informação a ser armazenada como índice é um número de 0 a 15.

IMAGENS RGB

- Imagens em RGB ou “true colors”, são imagens que para cada pixel são armazenadas as informações sobre as componentes, R(Red), G(Green) e B(Blue), da cor.
- O olho humano tem receptores para estas cores. Assim sendo, as demais cores são interpretadas pelo cérebro a partir dessas três cores primárias.

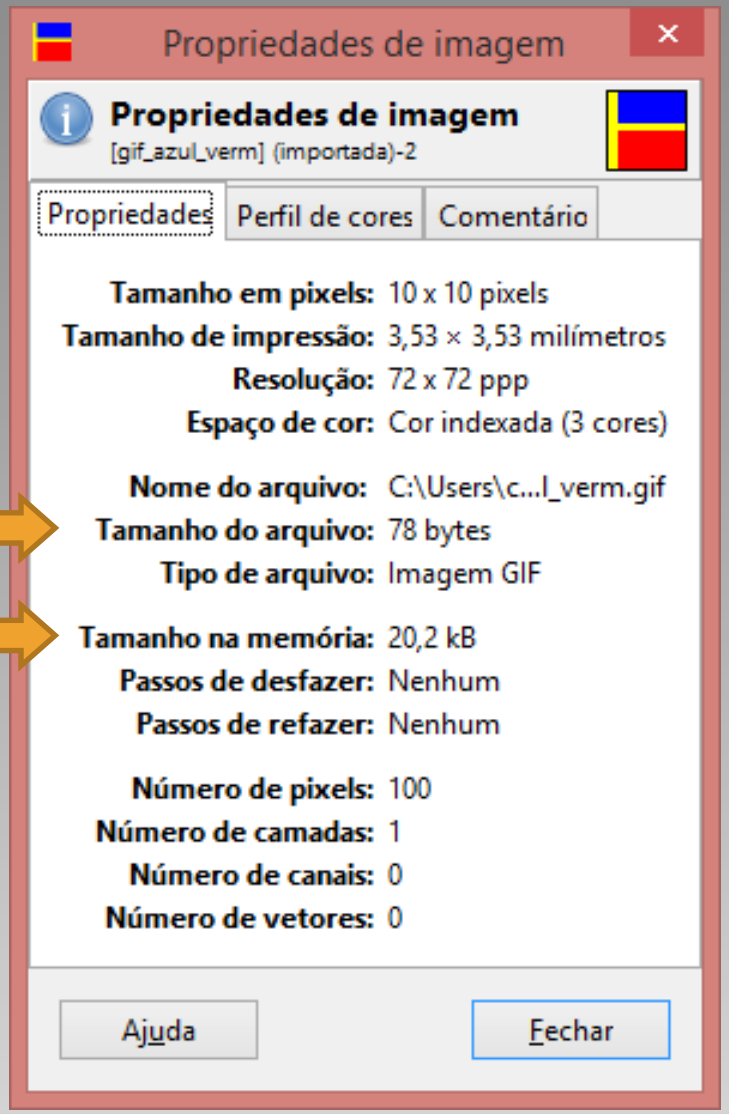
CONSUMO DE MEMÓRIA

- Uma imagem paletizada em 256 cores vai ocupar normalmente um valor fixo de cabeçalho, a paleta 256*4 Bytes e 1 Byte para cada pixel. Neste caso uma imagem de 128X128 ocupa em torno de 128x128x1 + 256*4 = 17408 Bytes.
- Uma imagem paletizada em 16 cores vai ocupar normalmente um valor fixo de cabeçalho, a paleta 16*4 Bytes e 1/2 Byte para cada pixel. Neste caso uma imagem de 128X128 ocupa em torno de 128x128x1/2 + 16*4 = 8256 Bytes.

CONSUMO DE MEMÓRIA

Observação: Normalmente as imagens que são carregadas na memória são expandidas para um formato em RGB compatível com o display em que serão desenhadas. Assim sendo, normalmente as imagens ocupam mais espaço(memória) quando estão na memória do que quando estão armazenadas (ex.: em disco).

CONSUMO DE MEMÓRIA



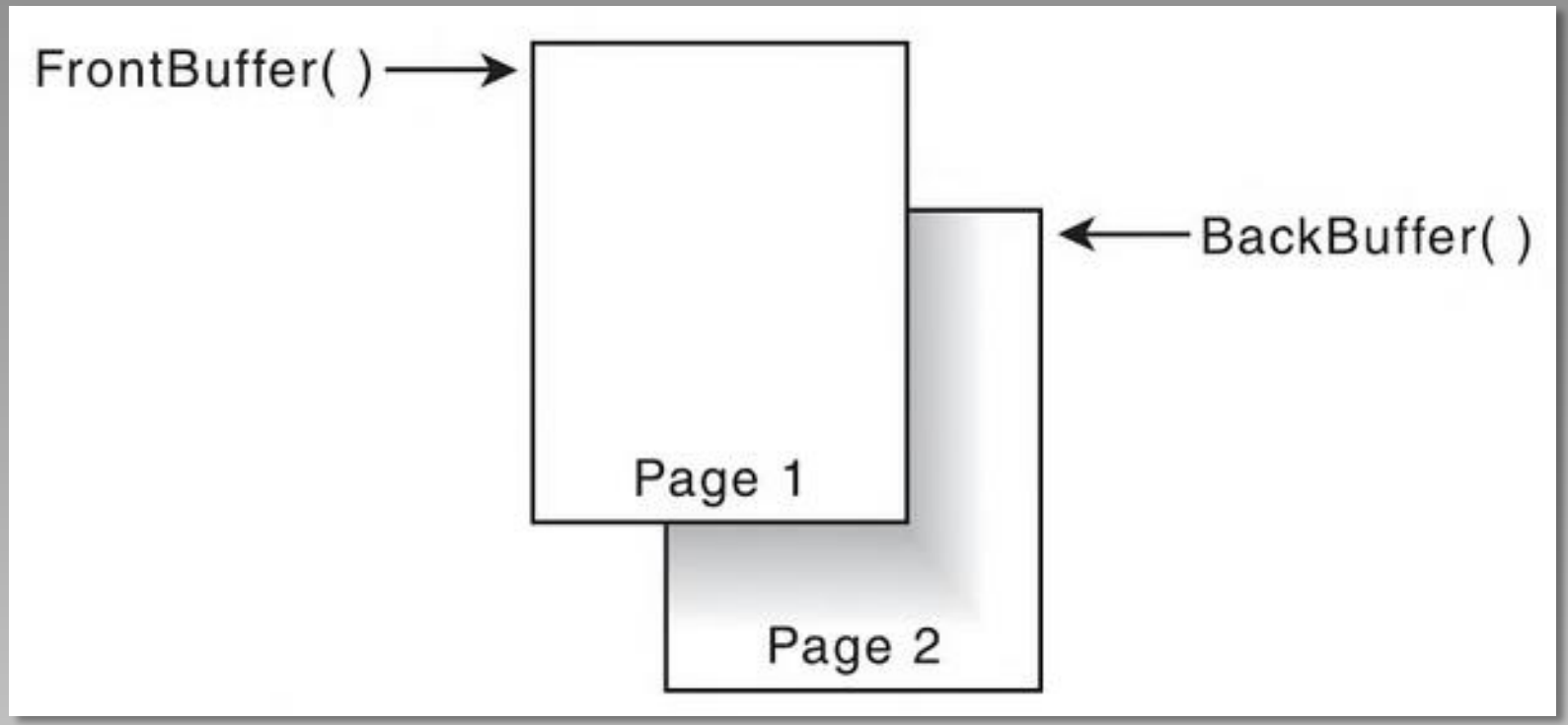
SUMÁRIO

- Tipos de imagem
- **Manipulação de imagens**
- Temporização
- Animações

BUFFERS DE IMAGENS

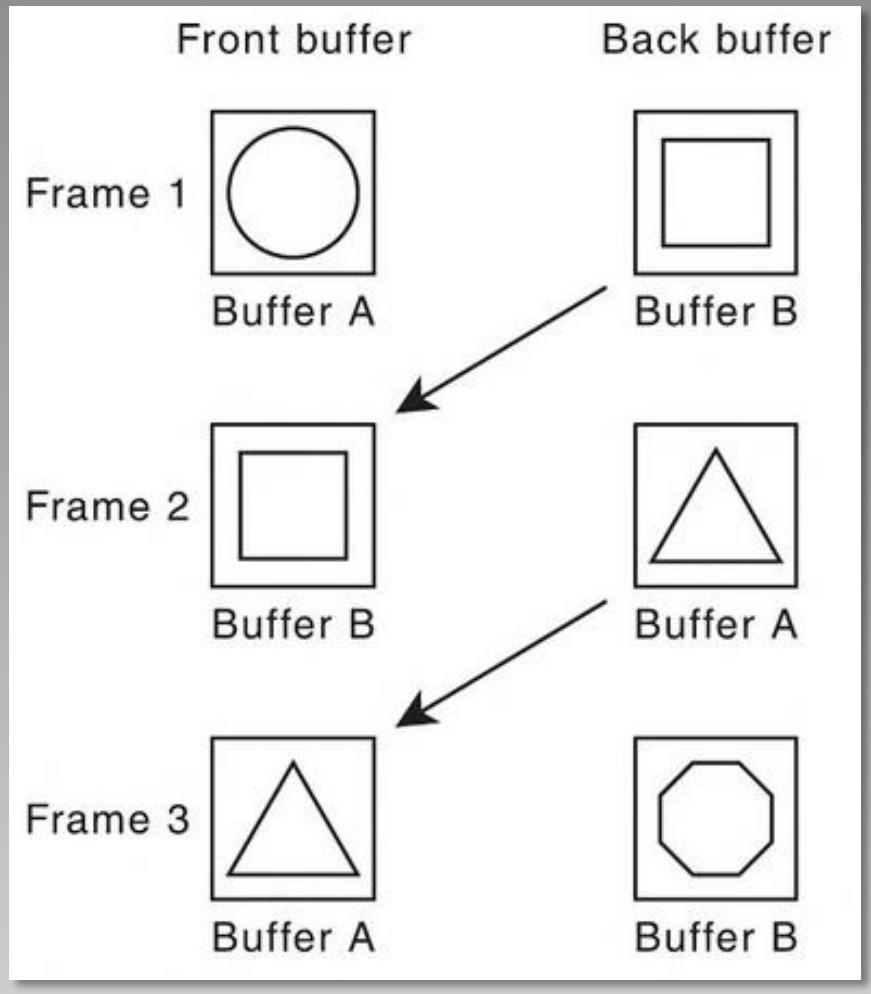
- Controle de sincronização entre os frames
- *Double Buffering*
- Buffer é uma imagem
- Cada frame do jogo é desenhado em um buffer
- **BackBuffer** é o buffer que está sendo desenhado
- **FrontBuffer** é o buffer que está sendo mostrado

BUFFERS DE IMAGENS



Fonte: Game Programming for Teens. 3rd Edition. Maneesh Sethi. Cengage Learning, 2009.

BUFFERS DE IMAGENS



Fonte: Game Programming for Teens. 3rd Edition. Maneesh Sethi. Cengage Learning, 2009.

BUFFERS DE IMAGENS

- BackBuffer e FrontBuffer devem ter o mesmo tamanho que a tela do jogo

BUFFERS DE IMAGENS

Exemplo

```
BufferedImage dbImage = new BufferedImage(
PWIDTH,HEIGHT,BufferedImage.TYPE_INT_ARGB);


dbImage.getGraphics().drawImage(...);
dbImage.getGraphics().fillRect(...);
public void paint(Graphics g) {
  g.drawImage(dbImage, 0, 0, null);
}
```

MANIPULAÇÃO DE IMAGENS

- Shaders: conjuntos de instruções que definem o comportamento de superfícies de objetos na tela. Ex.: reflexos na água de rios, porosidade de objetos e peles, movimentação da água com o personagem nadando, efeito especial de fogo.
- O objetivo é gerar aparências/efeitos mais realistas.

MANIPULAÇÃO DE IMAGENS

- Pixel Shader: atua diretamente no pixel, modificando padrões de cor e iluminação.

- http://pixelshaders.com/editor/

SUMÁRIO

- Tipos de imagem
- Manipulação de imagens
- **Temporização**
- Animações

TEMPORIZAÇÃO

- Para que os eventos, animações e movimentação ocorram na mesma velocidade, independente da plataforma em que o jogo está rodando, todo o jogo deve ser baseado em tempo.

TEMPORIZAÇÃO

- Num jogo em tempo real o mundo deve ser atualizado na mesma proporção em que o tempo está passando.
- Para que isso ocorra, a cada ciclo do loop principal deve-se calcular o tempo que o ciclo demorou para ser executado. Esse tempo é a diferença de tempo entre o final do ciclo atual e o final do ciclo anterior. Essa diferença de tempo será utilizada para atualizar o mundo no próximo ciclo.

TEMPORIZAÇÃO

```
while(jogo_rodando){

  processaEntradas();
  simulaJogo(difTempo);
  redesenha();

  difTempo = tempoAtual() - tempoFinalCicloAnt;
  tempoFinalCicloAnt = tempoaAtual();
}
```

FPS

- FPS ou *Frames per Second* é uma medida utilizada para quantidade de vezes que o ciclo do loop principal é executado por segundo.
- Essa é a medida de desempenho de um jogo em tempo real.
- O FPS ideal para um jogo dinâmico é algo em torno de 25 FPS. Mas dependendo do estilo do jogo 5 FPS já é aceitável.

FPS

```
while(jogo_Rodando){

   processaTeclas();
   simulaJogo(dif_tempo);
   redesenha();

   dif_tempo = tempoaAtual - tempoFinalCicloAnt;
   tempoFinalCicloAnt = tempoaAtual;
   FPS = 1000/dif_tempo;
}
```

FPS

```
while(jogo_rodando){
   processaEntradas();
   simulaJogo();
   redesenha();
}
```

25 ms

FPS = 1000/25

FPS = 40

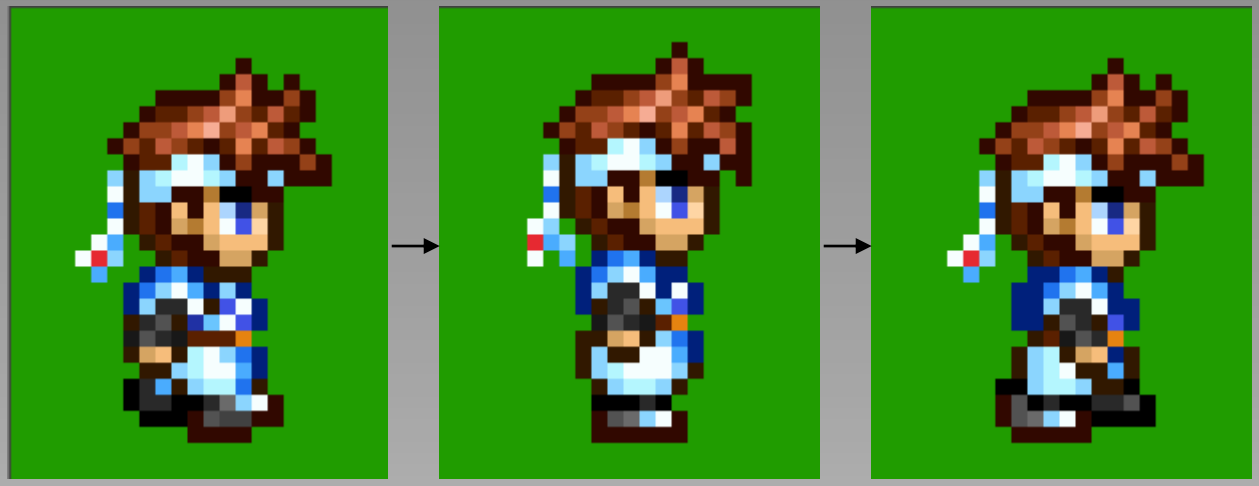
SUMÁRIO

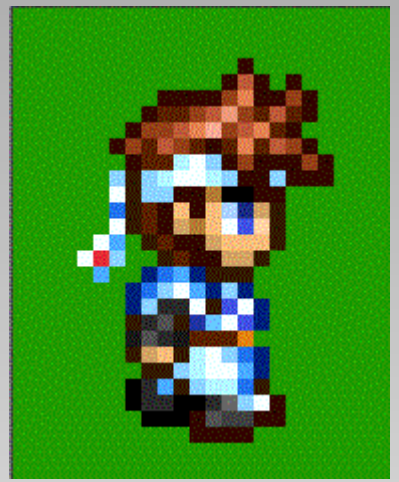
- Tipos de imagem
- Manipulação de imagens
- Temporização
- **Animações**

ANIMAÇÕES 2D

- Em jogos animações são criadas comumente através de **sequências de imagens**.
- Uma animação possui um número de imagens, frames, e uma temporização entre os frames.
- Uma animação pode ser cíclica ou não.

ANIMAÇÕES 2D





SPRITE

- Figura ou imagem criada a partir de um bloco de pixels, que pode ser movimentada na tela livremente, como uma unidade gráfica autônoma e totalmente independente das outras imagens de fundo.
- Este recurso é intensivamente utilizado em jogos e animações, constituindo-se numa forma de representação e movimentação rápida e controlada de objetos na tela.

SPRITE

SPRITE

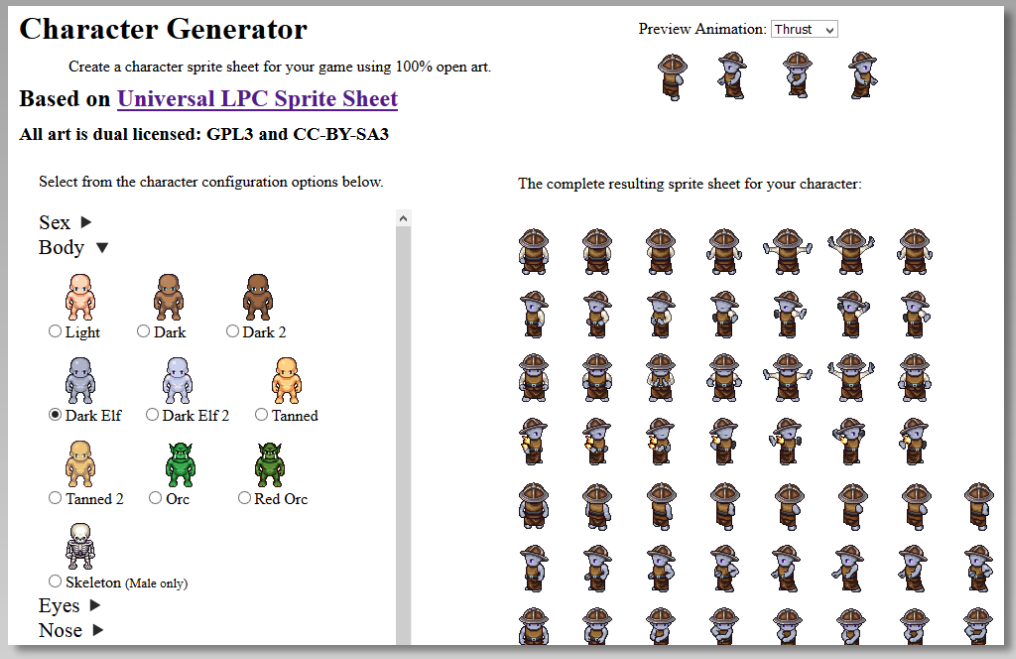
Site com vários *Sprite Sheets* de jogos conhecidos:

https://www.spriters-resource.com/

SPRITE

Gerador de *Sprite Sheets* para personagens:

http://gaurav.munjal.us/Universal-LPC-Spritesheet-Character-Generator/



OBJETOS

- Objetos são todos os atores de um jogo, podem ou não ser Sprites. Os objetos de um jogo devem ser atualizados (simulados) durante a etapa de simulação do jogo.
- Normalmente um objeto tem um método para simulação e um método de desenho que faz com que ele se desenhe no vídeo (buffer) com sua configuração atual.

LISTA DE OBJETOS

- A lista de objetos é uma lista que possui todos os objetos do jogo. Durante a etapa de simulação do jogo essa lista é percorrida e todos os objetos são simulados.

- Essa lista é percorrida novamente no momento do desenho para que todos os objetos visíveis sejam desenhados.

REDESENHO

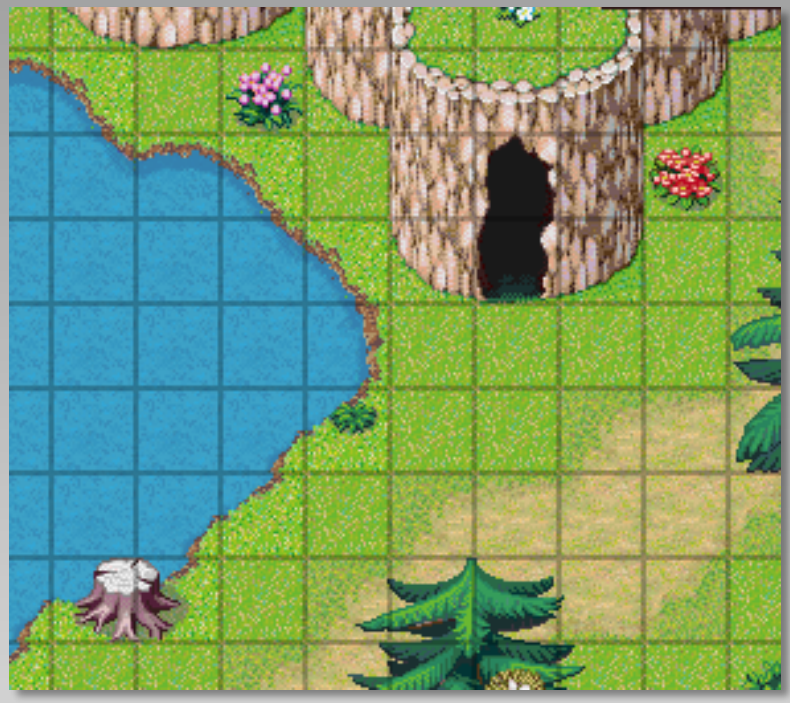
- Essa é a última etapa de um ciclo do loop principal. É nessa etapa que o mundo na sua configuração atual é desenhado na tela.
- Aqui todos os objetos pertencentes ao mundo e que estão visíveis são desenhados no vídeo. A posição onde serão desenhados depende de seus parâmetros.
- Todas as modificações realizadas no mundo na etapa de simulação já estão valendo.

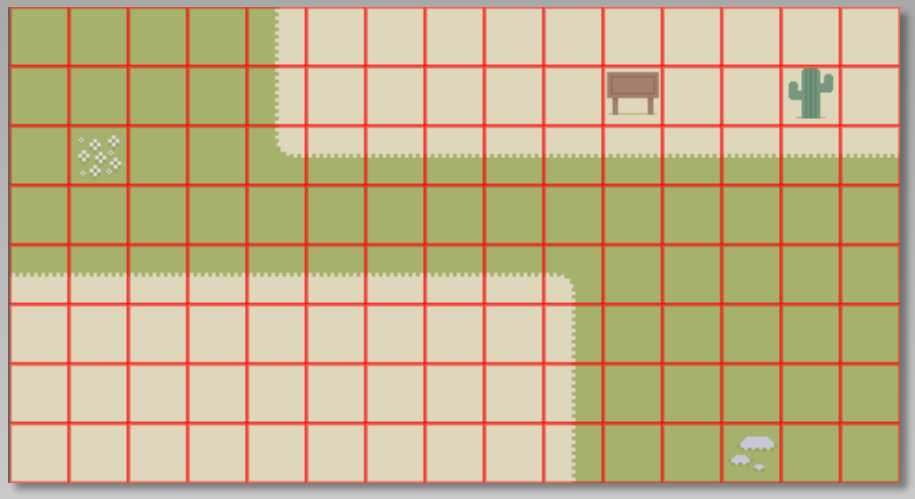
REDESENHO

```
redesenha(){
   limpaTela();
    desenhaImagem(
        imagem_personagem,
        Personagem.x,
        Personagem.y
   );
   ...
}
```

MANIPULAÇÃO DE IMAGENS

- Tilemap: mundo do jogo ou mapa de pequeno porte composto de imagens em forma de quadrado regular chamadas tile.





MANIPULAÇÃO DE IMAGENS

- Tileset: imagem contendo o conjunto dos tiles disponíveis para construir os Tilemaps.

MANIPULAÇÃO DE IMAGENS

- O uso de Tilemaps melhora a performance e o consumo de memória.

- http://mozdevs.github.io/gamedev-js-tiles/

MANIPULAÇÃO DE IMAGENS

- Tile Map Editor

http://sourceforge.net/projects/tilemapeditor2d

MANIPULAÇÃO DE IMAGENS