



Universidade
Tuiuti do
Paraná
Formação em essência

Disciplinas Optativas

Fundamentos de Jogos Digitais

Evandro Alberto Zatti

**Sprites Animados
(*spritesheets*)**



Sprite / Textura

- A exibição de uma imagem (Texture2D) em tela é feita através do método **Draw**, da classe **SpriteBatch**.
- O método Draw apresenta diversas sobrecargas, sendo que para a exibição de uma única imagem estática, pode-se utilizar sua chamada mais simples.
- Observe, a seguir, a exibição de uma imagem que foi vinculada ao gerenciador de conteúdo (**Content**), e carregada para um objeto **textura** do tipo Texture2D.





Sprite / Textura

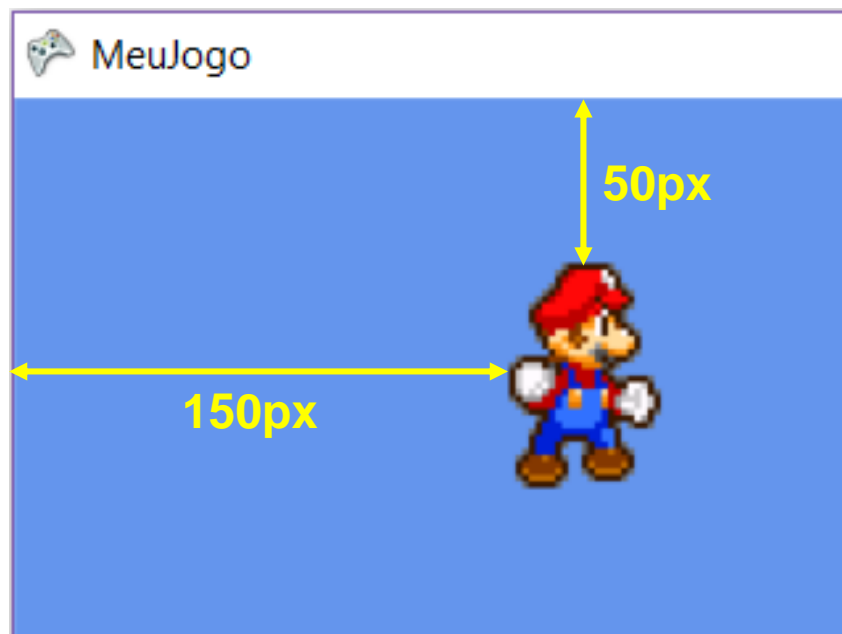
- O primeiro argumento para esta sobrecarga é a própria **textura** (Texture2D);
- O segundo argumento exige um **retângulo** (Rectangle), indicando a posição (X, Y) e o tamanho (largura, altura) da textura que será desenhada.
- O terceiro argumento é um filtro de **cor** aplicado à textura.

```
spriteBatch.Draw(  
    textura,  
    new Rectangle(150, 50, textura.Width, textura.Height),  
    Color.White  
);
```



Sprite / Textura

- Neste exemplo, a textura é desenhada a 150 pixels a partir da borda esquerda da janela, e a 50 pixels a partir do topo da janela. Foram utilizadas as propriedades Width e Height da própria textura para composição do tamanho do retângulo). A cor branca é utilizada para manter as cores originais da textura.





Sprite Animado














- A animação de um **sprite** é feita utilizando-se uma única imagem (.png), contendo todos os quadros (*frames*) para todos os estados do personagem:



- Este arquivo com os diversos quadros de animação é chamado de **spritesheet** (folha de sprites).
- Para facilitar a programação da animação, imagina-se a textura como sendo uma matriz de quadros, onde cada linha representa um estado do personagem e os quadros de cada estado são distribuídos nas colunas daquela linha, de forma equidistante.



Sprite Animado

	0	1	2	3	4	5	6	7	8	9
0										
1										

- A linha **0** (zero) da matriz apresenta os quadros referentes ao estado de **idle** (parado) do personagem; neste caso são 3 quadros (0 a 2).
- A linha **1** (um) da matriz apresenta os quadros referentes ao estado de **correndo** do personagem; para este estado, são 10 quadros (0 a 9).



Sprite Animado

- Para tanto, é importante criar uma variável de classe que faça o controle de qual célula da matriz de sprites deve ser exibida, isto é, qual o **frame** da animação será apresentado:

```
private Vector2 frame;
```

- No construtor da classe, inicia-se o frame com 0 (zero) para a linha (Y), indicando que seu estado inicial é parado, e também 0 (zero) para a coluna (X), para iniciar a exibição sucessiva a partir do primeiro quadro:

```
frame = new Vector2(0, 0);
```



Sprite Animado

- É importante também criar uma variável de classe que armazene o tamanho (largura e altura) de um único quadro, para ser utilizada na exibição:
`private Vector2 tamanho;`
- O tamanho deve ser pré-fixado no construtor:
`tamanho = new Vector2(35, 35);`
- Desta vez não será possível utilizar as propriedades Width e Height da textura, pois elas contêm a medida de todo o spritesheet.



Sprite Animado

- Para controlar o estado do personagem, cria-se um enumerador de estados e uma variável para armazenar o estado atual.

```
public enum Estados { Idle, Correndo }
```

```
private Estados estado;
```

- No construtor da classe, define-se o estado inicial:

```
estado = Estados.Idle;
```



Sprite Animado

- O método **Update** será responsável por controlar qual frame deverá ser exibido:

```
frame.X++;  
if (estado == Estados.Idle)  
{  
    if (frame.X > 2)  
        frame.X = 0;  
    frame.Y = 0;  
}  
else if (estado == Estados.Andando)  
{  
    if (frame.X > 9)  
        frame.X = 0;  
    frame.Y = 1;  
}
```



Sprite Animado

- O método **Draw** irá apresentar o frame correto, através de uma sobrecarga mais completa:

```
spriteBatch.Draw(  
    textura,  
    new Rectangle(  
        (int)posicao.X, (int)posicao.Y,  
        (int)tamanho.X, (int)tamanho.Y ),  
    new Rectangle(  
        (int)(frame.X * tamanho.X), (int)(frame.Y * tamanho.Y),  
        (int)tamanho.X, (int)tamanho.Y ),  
    Color.White,  
    0f,   
    Vector2.Zero,  
    SpriteEffects.None,  
    0  
);
```

Destino:
coordenadas
da tela

rotação (ângulo)

origem
(vértice)
da rotação

Origem:
coordenadas da
textura (célula)

layer (camada)