

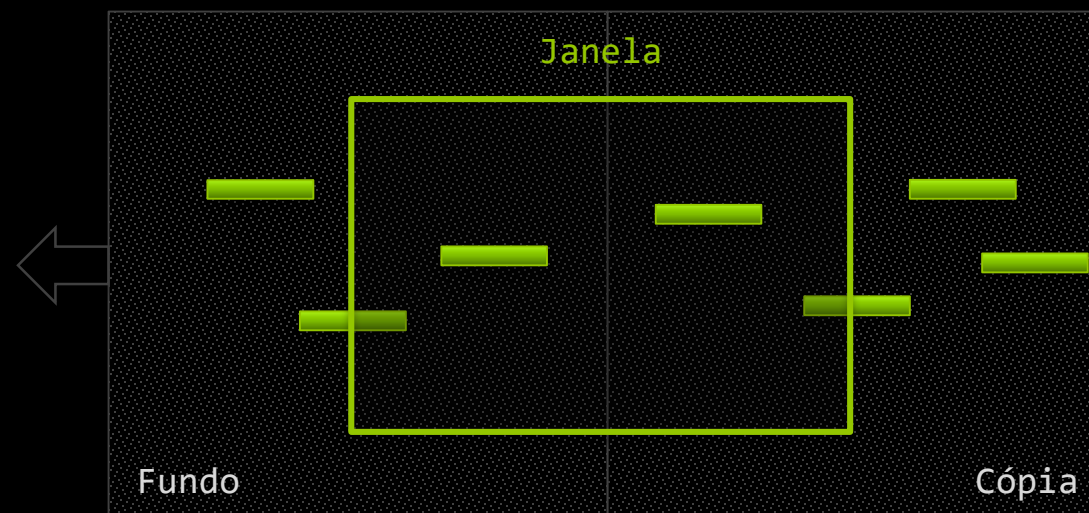
# Segundo Plano

## Programação de Jogos

Judson Santos Santiago

# Introdução

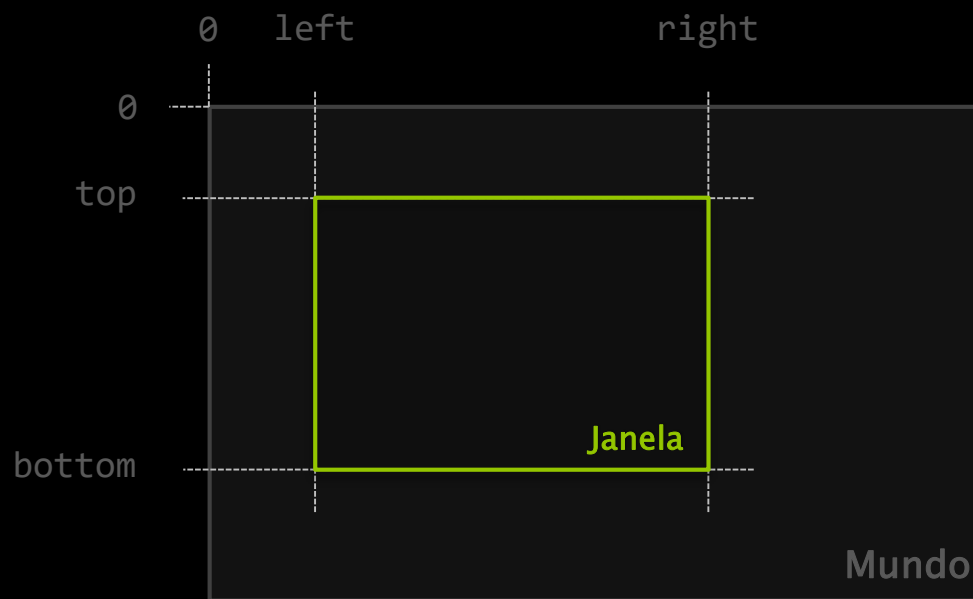
- ▶ Muitos jogos possuem mundos maiores que a tela
  - Esse efeito pode ser obtido através de:
    - Padrões repetitivos  
Ex.: Galaga
    - Carregamento dinâmico  
Ex.: Gravity Guy
    - Pano de fundo maior  
Ex.: Geometry Wars



# Viewport

- ▶ Ter um **mundo realmente maior** requer desenhar na área da janela apenas uma parte do mundo
  - Uma ViewPort pode ser usada para definir a área de exibição

```
struct ViewPort
{
    float left;
    float top;
    float right;
    float bottom;
};
```

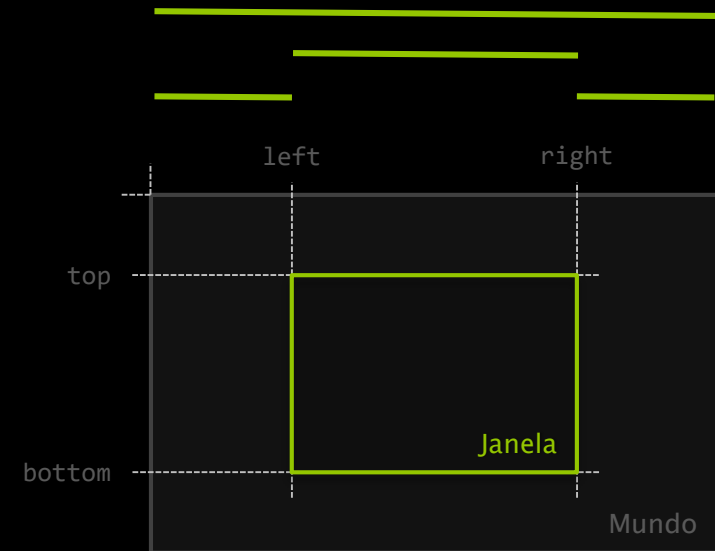


# ViewPort

## ► Podemos centralizar a viewport no mundo usando:

- O tamanho do pano de fundo\*
- O tamanho da janela

```
Background * backg;  
backg = new Background("Resources/BackgArt.jpg");  
  
// inicializa viewport para o centro do background  
viewport.left = (backg->Width() - window->Width()) / 2.0f;  
viewport.right = viewport.left + window->Width();  
viewport.top = (backg->Height() - window->Height()) / 2.0f;  
viewport.bottom = viewport.top + window->Height();  
  
// inicializa velocidade de rolamento da tela  
scrollSpeed = 400;
```



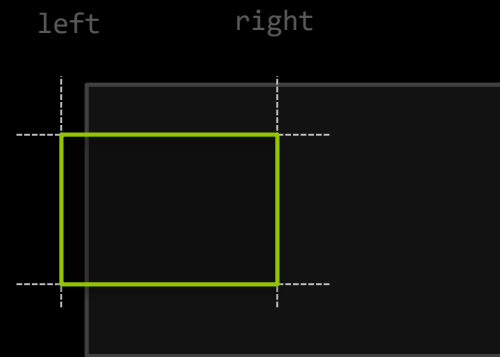
# ViewPort

- A cada frame a ViewPort deve ser ajustada

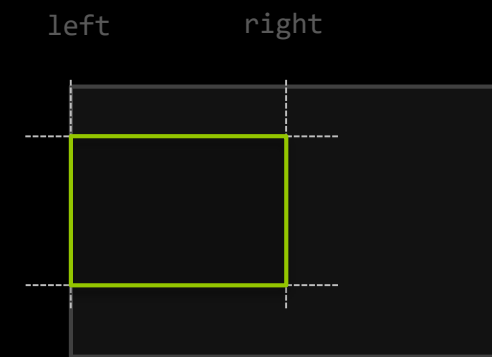
```
// deslocamento padrão
float delta = scrollSpeed * gameTime;

// movimenta viewport
if (window->KeyDown(VK_LEFT))
{
    viewport.left -= delta;
    viewport.right -= delta;

    if (viewport.left < 0)
    {
        viewport.left = 0;
        viewport.right = window->Width();
    }
}
```



ViewPort fora  
do mundo



ViewPort  
ajustada

# Mapas de Blocos

- ▶ A técnica de movimentação do segundo plano também pode ser aplicada em jogos cujo o mundo é formado por blocos

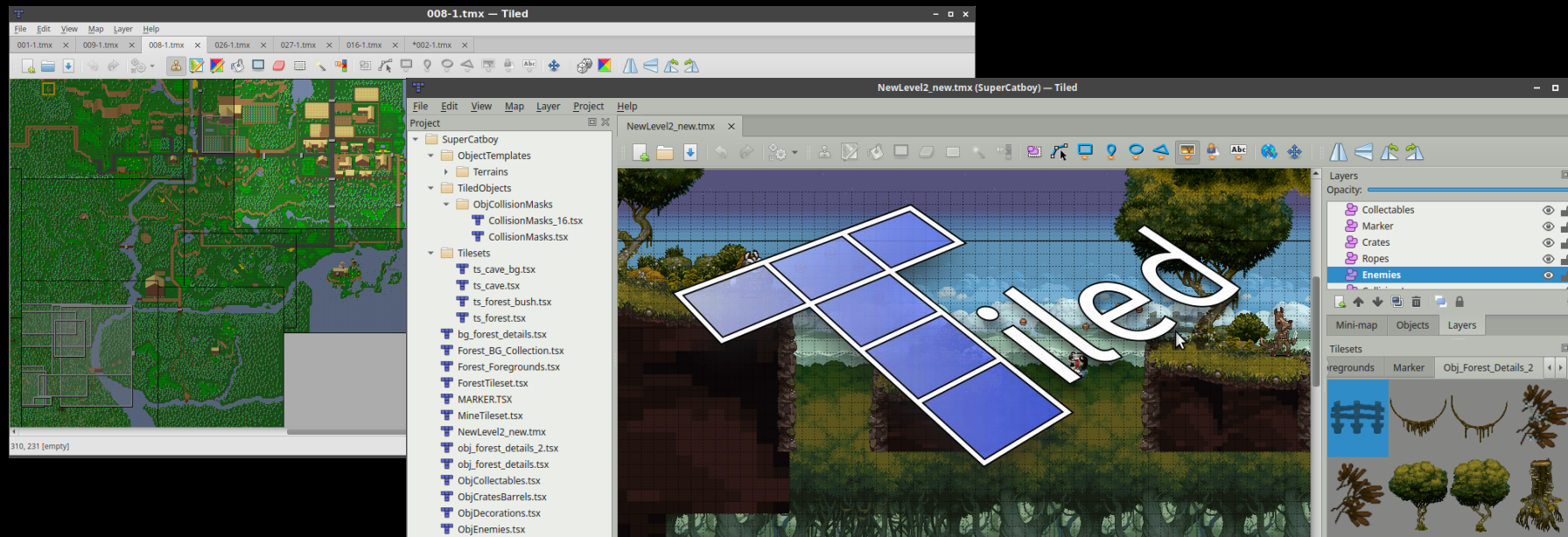


Alchemic Cutie



# Mapas de Blocos

- ▶ Estes mundos podem ser criados em editores especializados na criação de mapas



<https://www.mapeditor.org/>

# Mapas de Blocos

## ► A utilização de blocos requer **montagem**

### ◦ É preciso conhecer:

- **Blocos**

- Largura
- Altura
- Colunas

- **Mapa**

- Linhas
- Colunas

Blocos

1	2	3
4	5	6

numColumns

tileHeight



tileWidth

Mapa

tiledMapRows	1	1	2	1	1	1
	3	4	2	1	1	2
	1	1	2	1	1	5
	6	6	2	2	4	5
	6	6	2	2	1	3
	6	6	2	3	3	3

tiledMapCols



# Mapas de Blocos

## ► O mapa é armazenado como uma lista de índices

- Podem ser lidos para um vetor

0	1	2	3	4	5	6	7	8	9	10
1	1	2	1	1	4	4	2	1	3	...

- Acessados como uma matriz
- Os índices começam em 1

```
// bloco na posição [i][j]
uint tileNum = tiledMap[j + i * tiledMapCols] - 1;
```

```
// exemplo: bloco na posição [1][4]
tileNum = tiledMap[4 + 1 * 5] - 1
tileNum = 2
```

```
// mapa (linhas x colunas)
int tiledMap[6 * 5];
```

tiledMapRows	1	1	2	1	1
	4	4	2	1	3
	1	1	2	1	1
	6	6	2	2	4
	6	6	2	2	1
	6	6	2	2	1
tiledMapCols					

# Mapas de Blocos

## ► Os blocos são fornecidos em uma imagem

- Precisa ser recortada a partir de um índice

```
uint tileNum = tiledMap[i + j * tiledMapCols] - 1;
```

```
D3D11_BOX sourceRegion;
```

```
sourceRegion.left = (tileNum % numColumns) * tileWidth;
```

```
sourceRegion.top = (tileNum / numColumns) * tileHeight;
```

```
sourceRegion.right = sourceRegion.left + tileWidth;
```

```
sourceRegion.bottom = sourceRegion.top + tileHeight;
```

```
sourceRegion.front = 0;
```

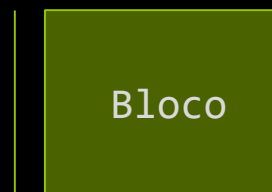
```
sourceRegion.back = 1;
```

Blocos

0 <del>1</del>	1 <del>2</del>	2 <del>3</del>
3 <del>4</del>	4 <del>5</del>	5 <del>6</del>

numColumns

tileHeight



tileWidth

# Mapas de Blocos

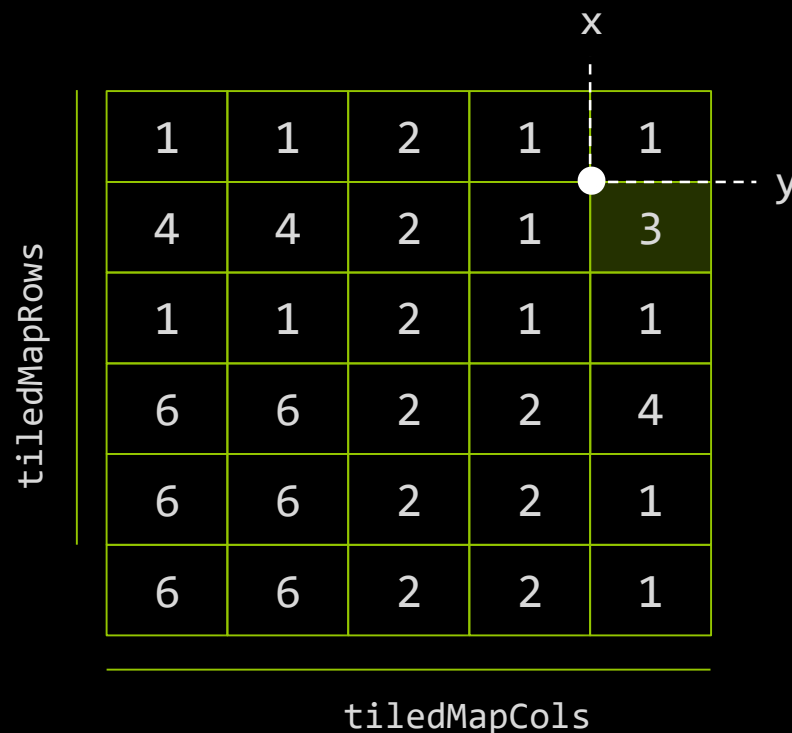
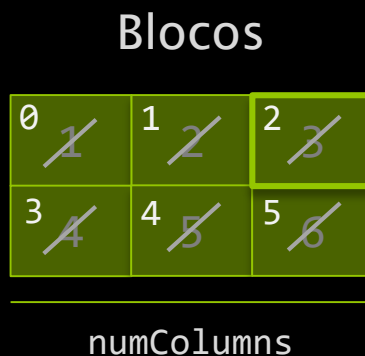
- Os blocos devem ser **copiados para uma imagem**

```
uint tileNum = tiledMap[i + j * tiledMapCols] - 1;
```

```
// coordenadas de destino
```

```
uint x = j * tileWidth;
```

```
uint y = i * tileHeight;
```



# Resumo

- ▶ O **segundo plano** ou **pano de fundo** tem uma função importante em muitos jogos
  - Mundo do jogo
  - Mapa
- ▶ A sua movimentação requer trabalhar com uma **superfície maior que a janela**:
  - Definição de uma ViewPort
  - Classe Background