



# Introdução à Engenharia

## ENG1000

### Aula 11 – Utilizando Imagens no Love2D 2016.1

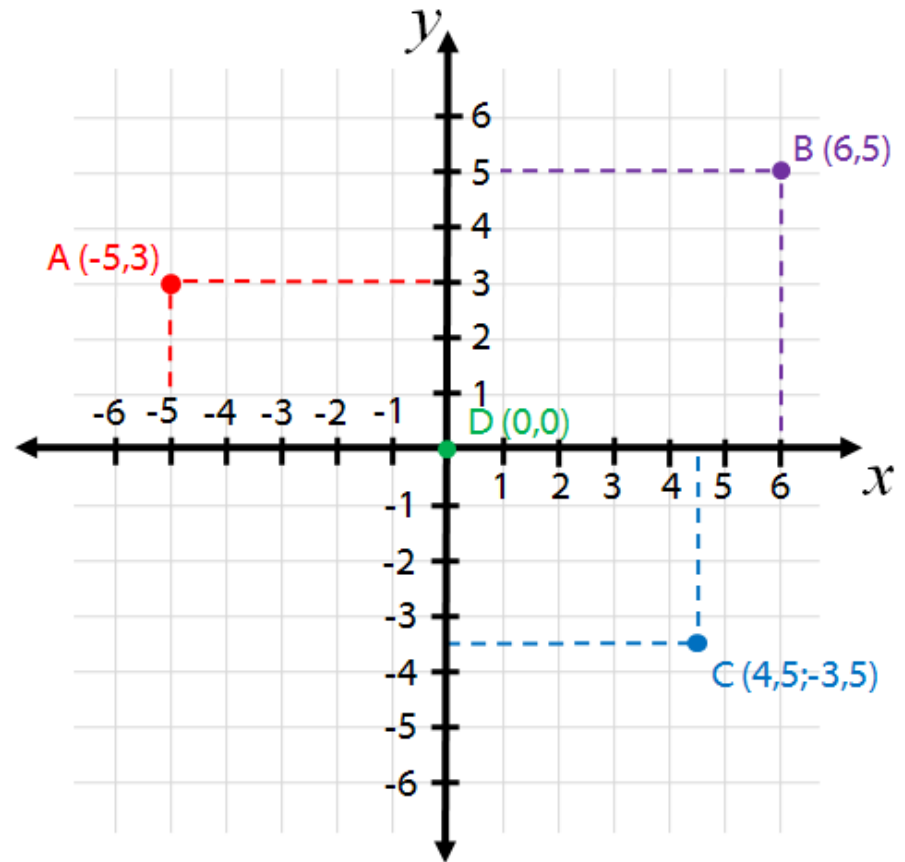


Prof. Augusto Baffa  
<[abaffa@inf.puc-rio.br](mailto:abaffa@inf.puc-rio.br)>

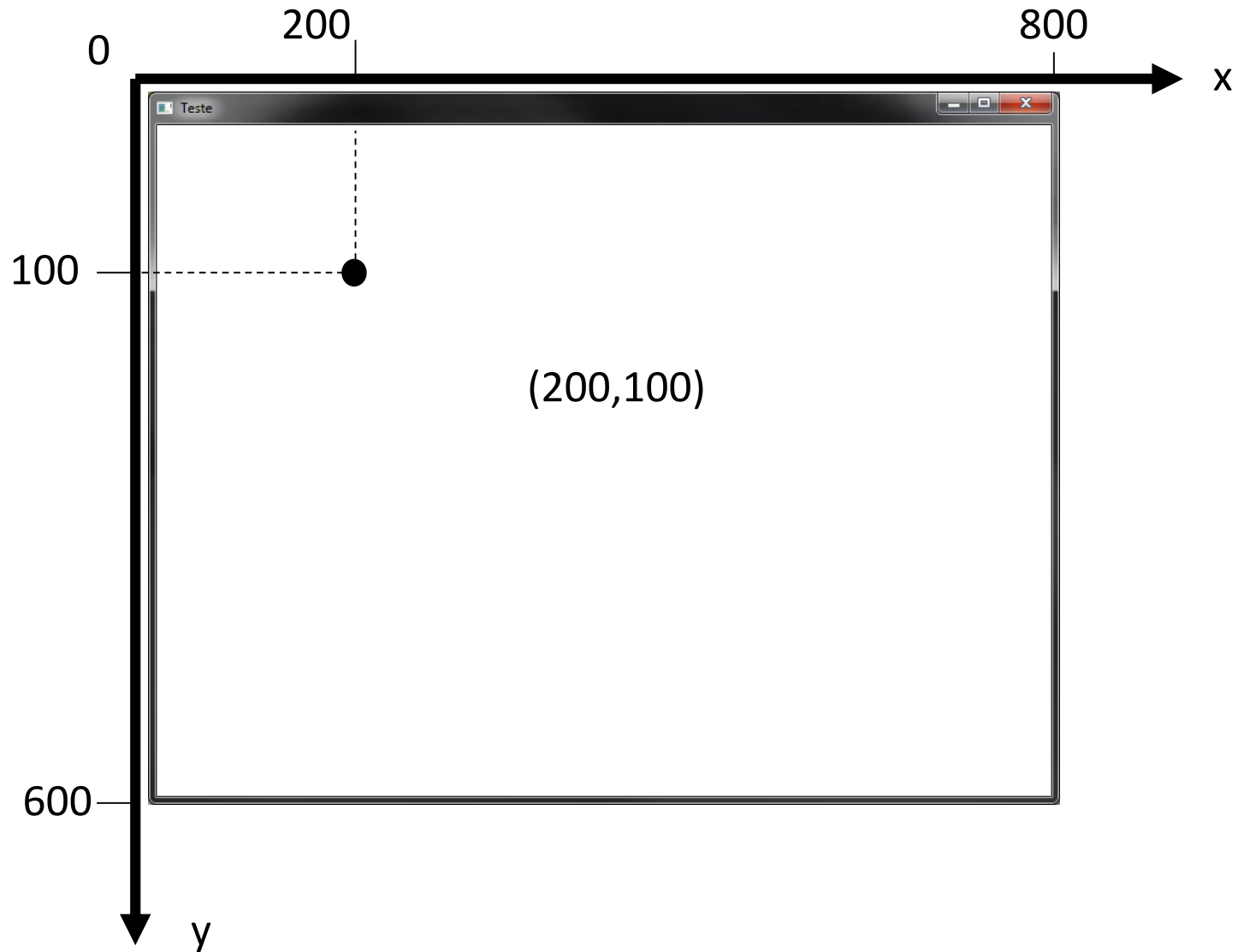


# Coordenadas de Tela

- Sistema de Coordenadas Cartesiano
- Duas dimensões (2D)
- Coordenas X e Y



# Coordenadas de Tela



# Desenhando Imagens

- **Para desenhar uma imagem na tela é necessário:**
  - **(1)** Criar uma variável para guardar o arquivo, que pode ser em qualquer formato (recomendado “.png”, em razão do fundo transparente), através do comando **love.graphics.newImage(“filename”)**.
  - **(2)** Desenhar efetivamente a imagem na tela usando o comando **love.graphics.draw(minha\_imagem, ...)**.

# love.graphics.newImage

- **(1) Criando uma variável do tipo image:**

```
minha_imagem
```

**OBS:** Sempre declare as variáveis Image como **variáveis globais**.

**Exemplo:**

```
function love.load()  
    minha_imagem = love.graphics.newImage("filename")  
end
```

# love.graphics.newImage

- **A variável do tipo image possui alguns métodos que podemos usar:**
  - `image.getHeight()` – Retorna o tamanho vertical da imagem (altura)
  - `image.getWidth()` – Retorna o tamanho horizontal da imagem (largura)

## **Exemplo:**

```
function love.load()  
    minha_imagem = love.graphics.newImage("filename")  
    imagem_width = minha_imagem.getWidth()  
    imagem_height = minha_imagem.getHeight()  
end
```

# love.graphics.draw

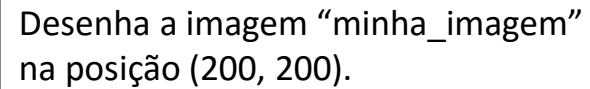
- **(2) Desenhar efetivamente a imagem na tela usando o comando:**

```
love.graphics.draw(image, x, y)
```

## Exemplo:

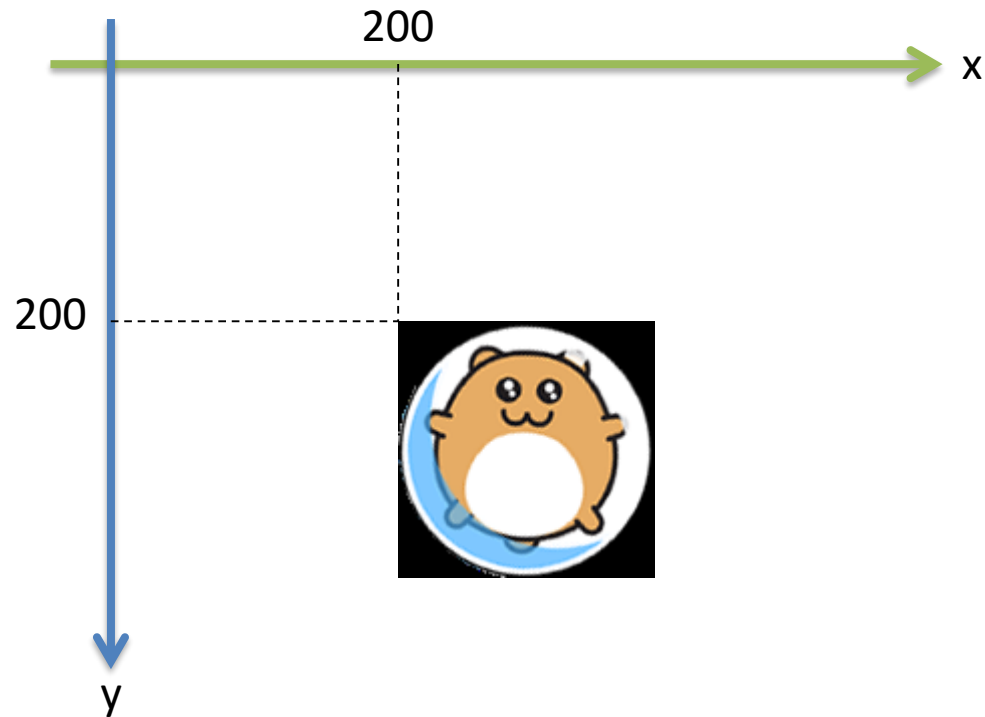
```
function love.draw()  
    (...)  
    love.graphics.draw(minha_imagem, 200, 200)  
    (...)  
end
```

Desenha a imagem "minha\_imagem"  
na posição (200, 200).



# Desenhando Imagens

- Por padrão, as imagens são desenhadas com o ponto de origem no canto superior esquerdo:





# love.graphics.draw

- **A função `love.graphics.draw` possui outros parâmetros que podem ser utilizados, por exemplo:**

```
love.graphics.draw(image, x, y, r, sx, sy, ox, oy, kx, ky)
```

- `image`: nome da variável associada no `load()`;
- `x`: posição onde o objeto será desenhado (eixo x);
- `y`: posição onde o objeto será desenhado (eixo y);
- `r`: orientação do objeto (radiano);
- `sx`: fator de escala do objeto (eixo x);
- `sy`: fator de escala do objeto (eixo y);
- `ox`: ponto de origem do objeto (eixo x);
- `oy`: ponto de origem do objeto (eixo y);
- `kx`: fator de distorção (eixo x); -- **Available since LÖVE [0.8.0](#)**
- `ky`: fator de distorção (eixo y); -- **Available since LÖVE [0.8.0](#)**

# love.graphics.draw

- Parâmetro `r`: orientação do objeto (radiano):
  - Realiza a rotação do objeto.



- Podemos definir uma velocidade angular para a rotação da imagem utilizando a função `update(dt)`.
- Para isto, inclua a expressão  $r = r + 20 * dt$ , fazendo com que a imagem gire de acordo com o tempo.

# love.graphics.draw

- Parâmetros  $sx$  e  $sy$ :
  - $sx$ : fator de escala do objeto (eixo x);
  - $sy$ : fator de escala do objeto (eixo y);



$sx = sy = 1$



$sx > 1$  e  $sy > 1$



$0 < sx < 1$   
 $0 < sy < 1$

# love.graphics.draw

- Parâmetros  $sx$  e  $sy$ :
  - E se o fator for  $sx < 0$  ou  $sy < 0$  ?

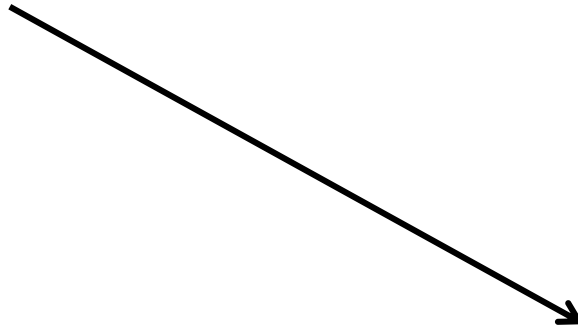


OU



# love.graphics.draw

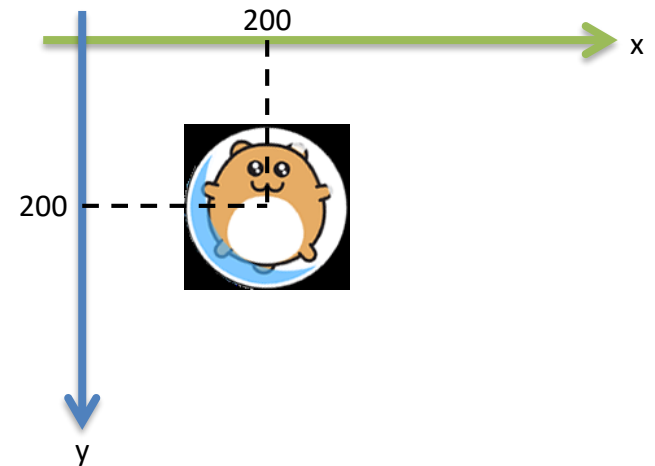
- Parâmetros  $sx$  e  $sy$ :
  - E se o fator for  $sx < 0$  e  $sy < 0$  juntos ?



# love.graphics.draw

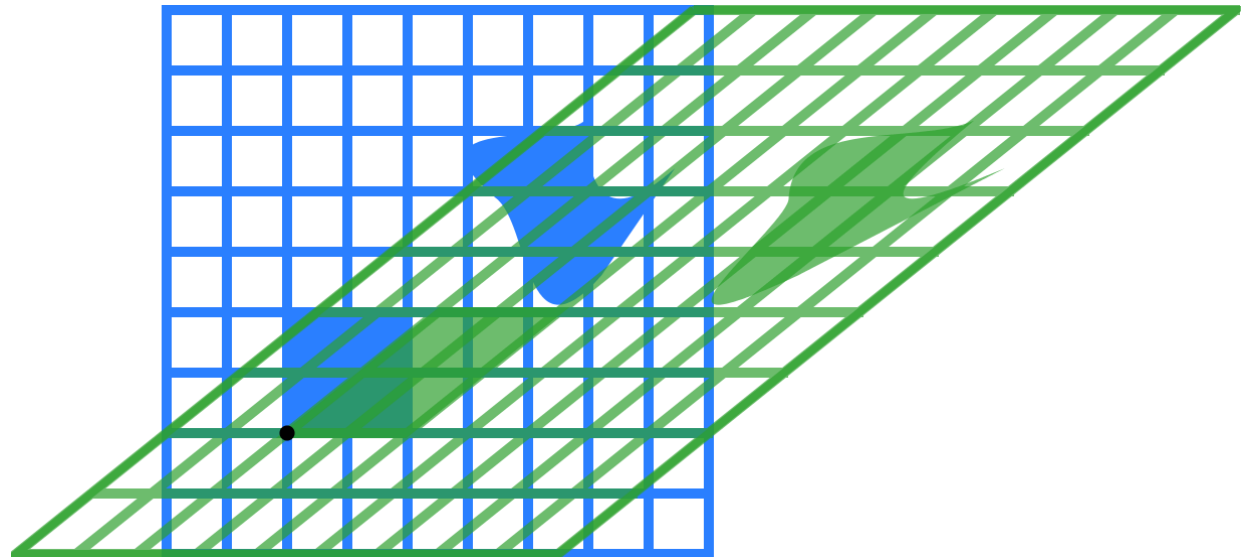
- Parâmetros `ox` e `oy`:
  - `ox`: ponto de origem do objeto (eixo x);
  - `oy`: ponto de origem do objeto (eixo y);
- Exemplo:

- `ox = minha_imagem:getWidth()/2`
- `oy = minha_imagem:getHeight()/2`



# love.graphics.draw

- Parâmetros  $k_x$  e  $k_y$ :
  - $k_x$ : fator de distorção (eixo x);
  - $k_y$ : fator de distorção (eixo y);



# Desenhando Imagens

- Desde a versão do LÖVE 0.9.0 foi incluído o parâmetro **quad** à função **draw**:

```
love.graphics.draw(image, quad, x, y, r, sx, sy, ox, oy, kx, ky)
```

- **Available since LÖVE [0.9.0](#)**
  - image: nome da variável associada no load();
  - **quads: vetor/matriz com imagens de um tileset (será melhor explicado posteriormente);**
  - x: posição onde o objeto será desenhado (eixo x);
  - y: posição onde o objeto será desenhado (eixo y);
  - r: orientação do objeto (radiano);
  - sx: fator de escala do objeto (eixo x);
  - sy: fator de escala do objeto (eixo y);
  - ox: ponto de origem do objeto (eixo x);
  - oy: ponto de origem do objeto (eixo y);
  - kx: fator de distorção (eixo x);
  - ky: fator de distorção (eixo y);



# Desenhando Imagens

- **Observações importantes sobre imagens:**
  - **Prefira utilizar imagens no formato PNG.** O formato PNG é um dos melhores formatos para esse tipo de aplicação. A principal vantagem é que ele permite o uso de **transparência** nas imagens.
  - Cerifique-se de que as imagens que serão lidas estão **dentro da pasta do seu projeto LÖVE2D**. Se preferir armazena-las em outras pastas você deve fornecer o caminho completo para o diretório onde as imagens estão para o comando `love.graphics.newImage`.
  - **Outros formatos** também são aceitos (JPG, GIF, BMP...). Em todo caso, é preferível que você converta sua imagem para o formato PNG antes de carregá-la.

# Exercícios

## **Lista de Exercícios 07 – Imagens e Interação**

<http://www.inf.puc-rio.br/~abaffa/eng1000/>