

# Introdução à Engenharia

## ENG1000

### Aula 10 – Introdução a PlayLib

Prof. Augusto Baffa

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

# Biblioteca Gráfica - PlayLib

- **Conjunto de funções** para criação e manipulação de formas geométricas, imagens, áudio, janelas...
- Baseada na API **OpenGL**.
- Pode ser usada para criação de **jogos 2D, simulações, animações** e outros aplicativos.
- **Desenvolvida especialmente para esse curso!**

# Estrutura de um Programa

```
#include "Graphics.h"
```

Inclusão da Biblioteca gráfica

```
Graphics graphics;
```

Instância de um objeto do tipo **Graphics** que permite o uso das funções gráficas.

```
int main(void)
```

```
{
```

```
    graphics.CreateMainWindow(800, 600, "Teste");
```

Cria uma **janela** de tamanho 800x600 com o título "Teste"

```
    graphics.StartMainLoop();
```

Inicia o **Loop principal** do programa

```
    return 0;
```

```
}
```

# Resultado do Programa Anterior?



# Loop Principal

- O **Loop Principal** consiste de uma função que é repetida enquanto o programa não for fechado pelo usuário.
- Todo processamento realizado pelo programa gráfico está de alguma forma ligado ao Loop Principal.
- **No Loop Principal deve ser programado:**
  - Os objetos que serão desenhados na tela e como eles serão apresentados;
  - Quais animações e movimentos os objetos terão.
  - Toda a lógica do programa.

# Definindo um Loop Principal

```
void MainLoop() ←  
{  
    graphics.SetColor(0,255,0); ←  
  
    graphics.FillRectangle2D(100, 100, 400, 200); ←  
}
```

Função que será usada como **Loop Principal** do programa

Define a **cor** que será utilizada para desenhar objetos na tela (Formato RGB)

Desenha um **retângulo** preenchido iniciando na posição (100,100) e indo até (200,400)

```
int main(void)  
{  
    graphics.CreateMainWindow(800, 600, "Teste");  
  
    graphics.SetMainLoop(MainLoop); ←  
  
    graphics.StartMainLoop();  
  
    return 0;  
}
```

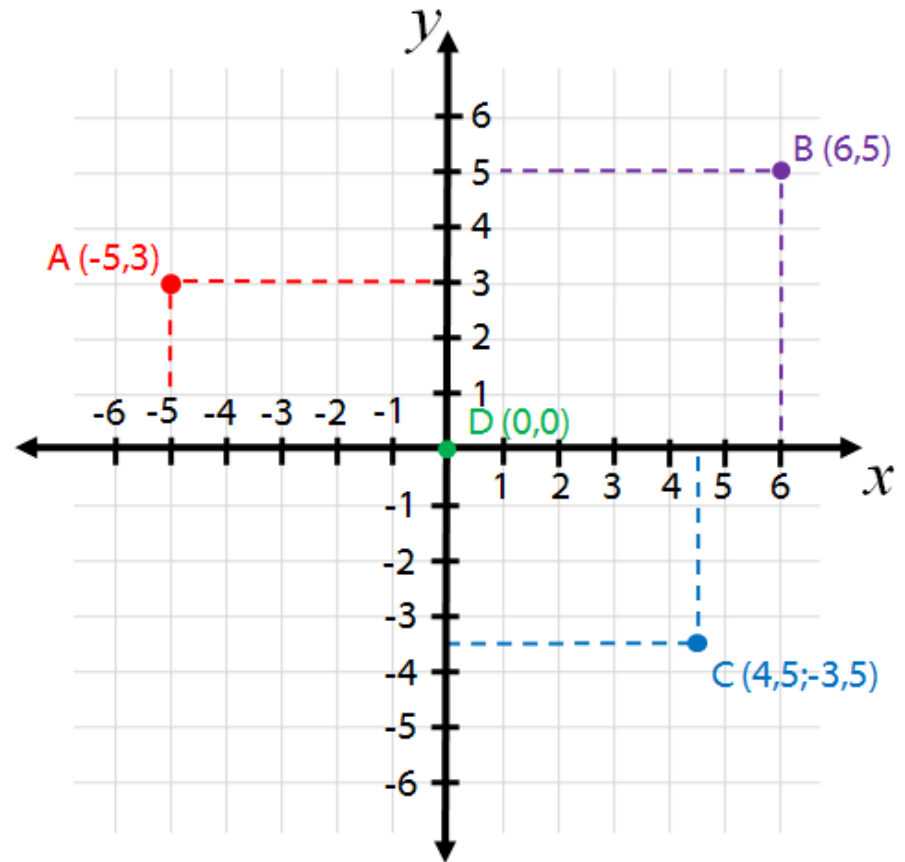
Define que a função **MainLoop** será o **Loop Principal** do programa

# Resultado do Programa Anterior?



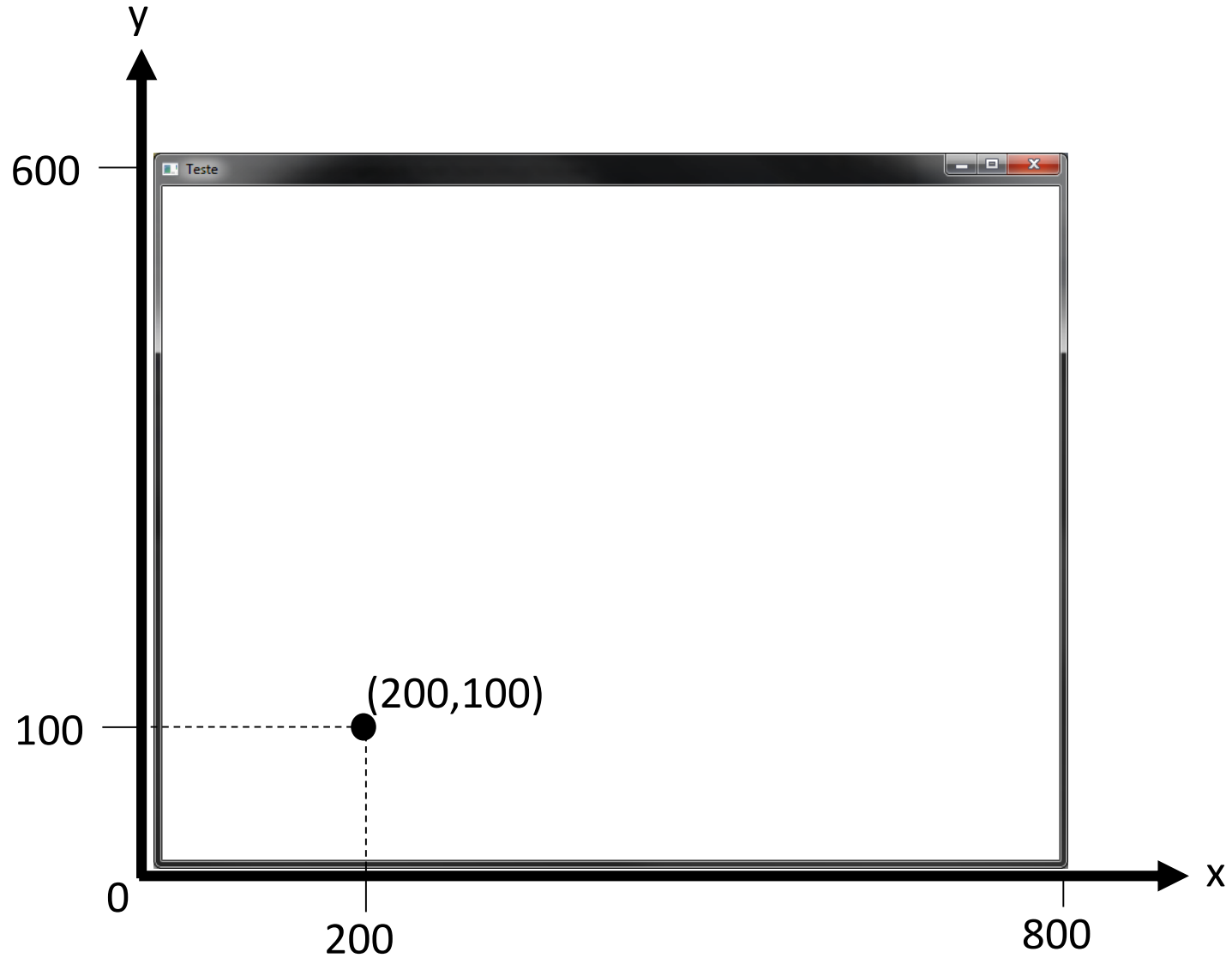
# Coordenadas de Tela

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





# Coordenadas de Tela



# Desenho de Primitivas Geométricas

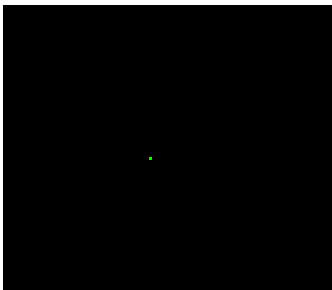
- **Ponto:**

```
void DrawPoint2D(int x, int y);
```

**Exemplo:**

```
graphics.DrawPoint2D(200, 200);
```

Desenha um ponto na posição  
(200, 200) da tela



# Desenho de Primitivas Geométricas

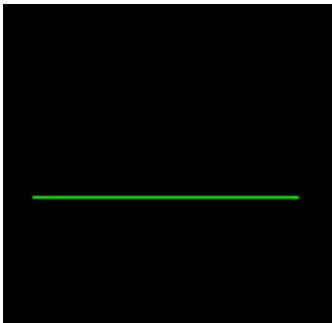
- **Linha:**

```
void DrawLine2D(int x1, int y1, int x2, int y2);
```

**Exemplo:**

```
graphics.DrawLine2D(100, 100, 200, 100);
```

Desenha uma linha saindo da posição (100, 100) e indo até a posição (200, 100)



# Desenho de Primitivas Geométricas

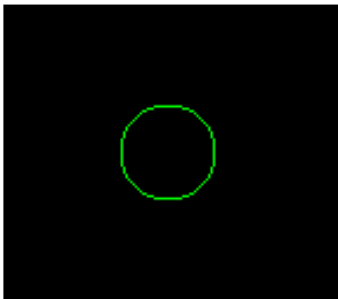
- **Círculo:**

```
void DrawCircle2D(int x, int y, int radius);
```

**Exemplo:**

```
graphics.DrawCircle2D(200, 200, 20);
```

Desenha um círculo de raio 20 na posição (200, 200) da tela



# Desenho de Primitivas Geométricas

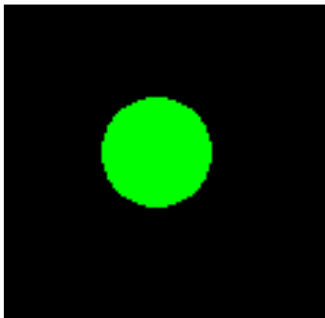
- **Círculo Preenchido:**

```
void FillCircle2D(int x, int y, int radius, int segments);
```

**Exemplo:**

```
graphics.FillCircle2D(200, 200, 20, 30);
```

Desenha um círculo preenchido de raio 20 com 30 segmentos na posição (200, 200) da tela. Quanto mais segmentos, mais redondo o círculo será.



# Desenho de Primitivas Geométricas

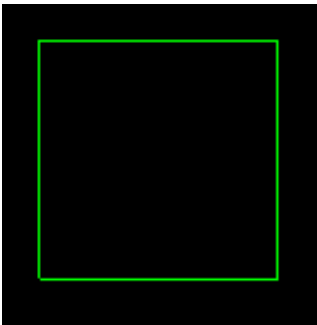
- **Retângulo:**

```
void DrawRectangle2D(int x1, int y1, int x2, int y2);
```

**Exemplo:**

```
graphics.DrawRectangle2D(100,100,200,200);
```

Desenha um retângulo iniciando na posição (100, 100) e indo até a posição (200, 200)



# Desenho de Primitivas Geométricas

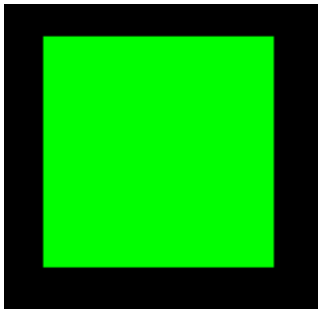
- **Retângulo Preenchido:**

```
void FillRectangle2D(int x1, int y1, int x2, int y2);
```

**Exemplo:**

```
graphics.FillRectangle2D(100,100,200,200);
```

Desenha um retângulo preenchido iniciando na posição (100, 100) e indo até a posição (200, 200)



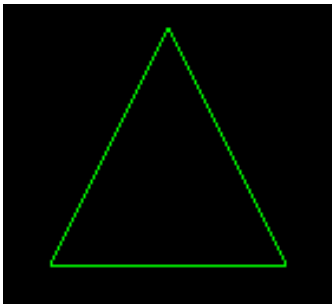
# Desenho de Primitivas Geométricas

- **Triângulo:**

```
void DrawTriangle2D(int x1, int y1, int x2, int y2, int x3, int y3);
```

**Exemplo:**

```
graphics.DrawTriangle2D(100,100,200,100,150,200);
```



Desenha um triângulo com o primeiro ponto na posição (100, 100), segundo ponto na posição (200, 100) e terceiro ponto na posição (150, 200)



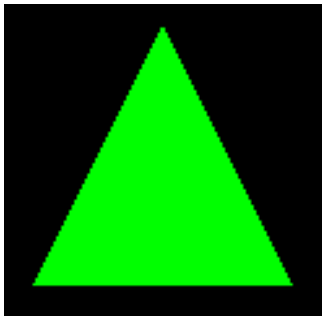
# Desenho de Primitivas Geométricas

- **Triângulo Preenchido:**

```
void FillTriangle2D(int x1, int y1, int x2, int y2, int x3, int y3);
```

**Exemplo:**

```
graphics.FillTriangle2D(100,100,200,100,150,200);
```



Desenha um triângulo preenchido com o primeiro ponto na posição (100, 100), segundo ponto na posição (200, 100) e terceiro ponto na posição (150, 200)

# Desenho de Primitivas Geométricas

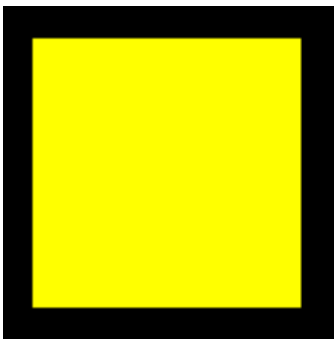
- **Modificando a Cor:**

```
void SetColor(float r, float g, float b);
```

**Exemplo:**

```
graphics.SetColor(255, 255, 0);
```

Altera a cor que será usada para desenhar os objetos para o valor RGB (255,255,0). Ou seja, mistura o máximo de vermelho com o máximo de verde, o que resulta em amarelo.



# Desenho de Primitivas Geométricas

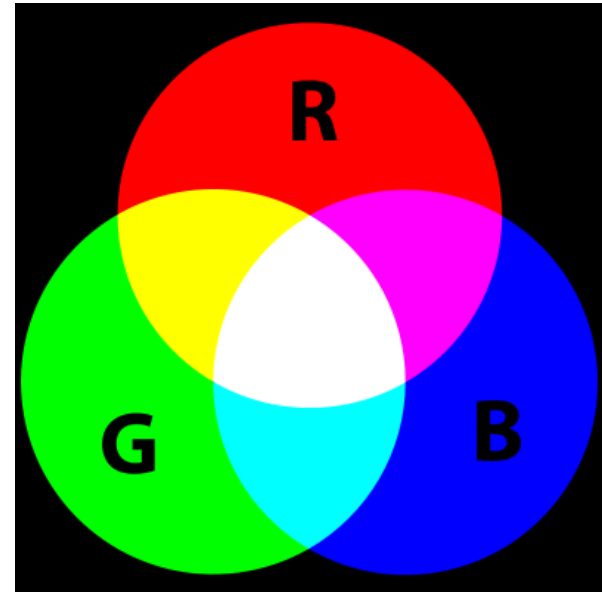
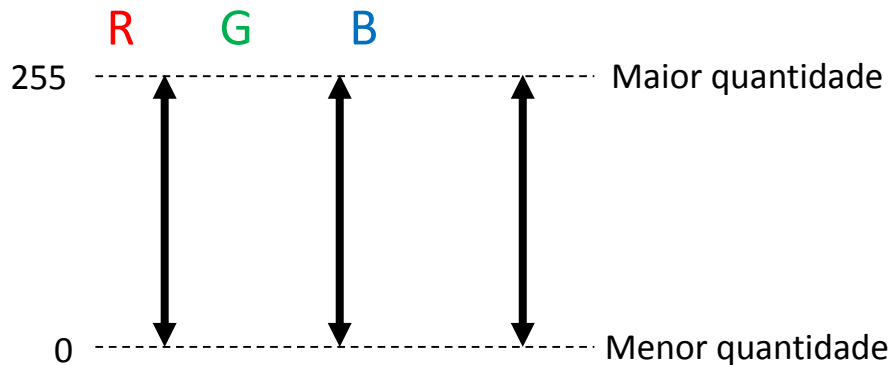
- **Formato de cor RGB:**

**R** = Red (Vermelho)

**G** = Green (Verde)

**B** = Blue (Azul)

- **Escala RGB:**



Não sabe o valor RGB da cor que  
você quer?

<http://www.colorpicker.com/>

# Desenho de Primitivas Geométricas

- **Modificando a Cor do Fundo da Tela:**

```
void SetBackgroundColor(float r, float g, float b);
```

**Exemplo:**

```
graphics.SetBackgroundColor(255, 255, 255);
```



Altera a cor do fundo da tela para o valor RGB (255,255,255). Ou seja, mistura o máximo de todas as cores, o que resulta em branco.

# Desenho de Primitivas Geométricas

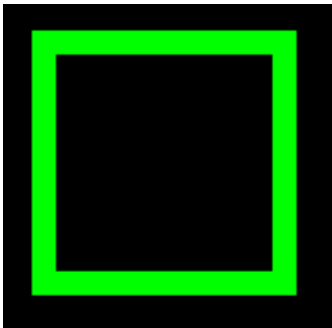
- **Modificando a Largura das Linhas:**

```
void SetLineWidth(float width);
```

**Exemplo:**

```
graphics.SetLineWidth(12);
```

Altera para 12 a largura das linhas usadas para desenhar as formas geométricas.



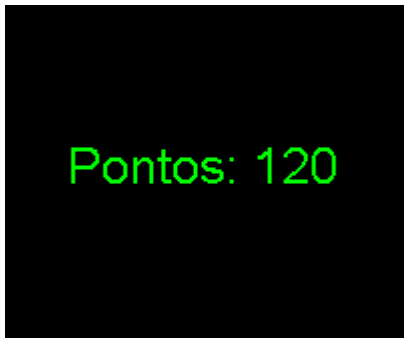
# Desenho de Primitivas Geométricas

- **Escrevendo um Texto na Tela:**

```
void DrawText2D(int x, int y, const char *text, ...);
```

**Exemplo:**

```
graphics.DrawText2D(100, 100, "Pontos: %d", pontos);
```



Escreve "Pontos: " seguido do valor da variáveis "pontos" na posição (100, 100) da tela.

# Desenho de Primitivas Geométricas

- **Modificando a Fonte do Texto:**

```
void SetTextFont(const char *font_name, int size, int weight,  
                bool italic, bool underline);
```

**Exemplo:**

```
graphics.SetTextFont("Times New Roman", 42, FONT_WEIGHT_BOLD,  
                    false, false);
```



**Pontos: 120**

Altera a fonte utilizada no programa para "Times New Roman", com tamanho 42, bold.

# Outras Funções

- **Criar a Janela do Programa:**

```
void CreateMainWindow(int sizeX, int sizeY, char title[]);
```

## Exemplo:

```
graphics.CreateMainWindow(800, 600, "Nome da Janela");
```



Cria a janela principal do programa com o tamanho 800x600 e com o título "Nome da Janela"



# Outras Funções


- **Executando o programa em tela cheia:**

```
void SetFullscreen(bool enable);
```

## **Exemplo:**

```
graphics.SetFullscreen(true);
```

Coloca o programa em tela cheia



```
graphics.SetFullscreen(false);
```

Remove o programa da tela cheia



# Outras Funções


- **Verificando a Velocidade de Execução do Programa:**

```
float GetFPS();
```

## Exemplo:

```
fps = graphics.GetFPS();
```

Coloca o número de frames por segundo na variável fps



- **FPS (Frames per Second):** Medida que nos indica quantos frames (imagens) consecutivos a placa de vídeo está conseguindo gerar por segundo.

# Outras Funções

- **Verificando a Velocidade de Execução do Programa:**

```
float GetElapsedTime();
```

## Exemplo:

```
PosicaoX = PosicaoX + (Speed * graphics.GetElapsedTime());
```



Calcula o deslocamento em X de forma independente da taxa de frames por segundo. Isso é muito importante, pois permite que a velocidade do deslocamento seja independente da velocidade que o jogo está sendo executado.

# Outras Funções

- **Verificando a Largura e a Altura da Tela:**

```
int GetScreenWidth();
```

```
int GetScreenHeight();
```

## **Exemplo:**

```
width = graphics.GetScreenWidth();
```

Coloca a largura da tela na  
variável width

```
height = graphics.GetScreenHeight();
```

Coloca a altura da tela na variável  
height

# Exercícios

## Lista de Exercícios 06 – Primitivas Geométricas

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

Enviar para [abaffa@inf.puc-rio.br](mailto:abaffa@inf.puc-rio.br) até 08/abril

Assunto: ENG01000 Ex06

Não Esquecer de colocar nome e matrícula.