

**Pontifícia Universidade Católica de Minas Gerais**

**Computação Gráfica**

**Profa. Rosilane Mota**

**Roteiro 01   
Transformações Geométricas**

De modo a padronizar uma IDE com manipulação por eventos de unidades gráficas e antecipar conhecimento para uso da Unity 3D, utilizaremos o Visual Studio Community 2019 com C# disponível no Portal Azure (Parceria ICEI). Outras versões ou IDEs podem ser utilizadas, desde que nas entregas seja incluído instalador para Windows.

1. Criar um novo projeto

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

1. Abrir e fixar caixa de ferramentas

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Sugestões de componentes e alterações em suas propriedades (coloque-as em ordem alfabética )  
   1. **Form**  
      (Name) tela  
      Text Roteiro 1 – Ambientação e Transformações   
       Geométricas
   2. **Panel**  
      (Name) painel

Dock Right

* 1. **PicureBox**

(Name) imagem

BackColor White

Dock Fill

Image nenhum

* 1. **Label – 2 deles**  
     (Name) lbX

Font inclua Negrito e aumente tamanho (cx diálogo **...**)  
Text X  
  
(Name) lbY

Font inclua Negrito e aumente tamanho (cx diálogo **...**)  
Text Y

* 1. **TextBox – 2 deles**  
     (Name) txtX  
     Text 0  
       
     (Name) txtY  
     Text 0
  2. **Button – 3 deles**  
     (Name) btDesenhar  
     Text Desenhar  
       
     (Name) btCor  
     Text Cor  
       
     (Name) btApagar  
     Text Apagar
  3. **ColorDialog**  
     (Name) cdlg

1. Crie dois atributos para a classe tela (Clique no formulário e pressione F7)  
   Bitmap areaDesenho;

Color corPreenche;

1. Inicialize os atributos no construtor da classe

public tela()

{

InitializeComponent();

areaDesenho = new Bitmap(imagem.Size.Width,   
 imagem.Size.Height);

corPreenche = Color.Black;

}

1. Criação e implementação de eventos (Clique 2x no botão correspondente)
   1. btDesenhar  
        
      private void desenhar\_Click(object sender, EventArgs e)

{

int x = (int) Convert.ToInt64(txtX.Text);

int y = (int) Convert.ToInt64(txtY.Text);

areaDesenho.SetPixel(x, y, corPreenche);

imagem.Image = areaDesenho;

}

* 1. btCor  
      private void btCor\_Click(object sender, EventArgs e)

{

DialogResult result = cdlg.ShowDialog();

if (result == DialogResult.OK)

{

corPreenche = cdlg.Color;

}

}

* 1. btApagar  
       
     private void btApagar\_Click(object sender, EventArgs e)

{

areaDesenho = new Bitmap(imagem.Size.Width,   
 imagem.Size.Height);

imagem.Image = areaDesenho;

}

* 1. imagem 

Inclua clicando 2x o evento MouseMove  
  
private void imagem\_MouseMove(object sender,   
 MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

int x = e.X;

int y = e.Y;

txtX.Text = Convert.ToString(x);

txtY.Text = Convert.ToString(y);

areaDesenho.SetPixel(x, y, corPreenche);

imagem.Image = areaDesenho;

}

}