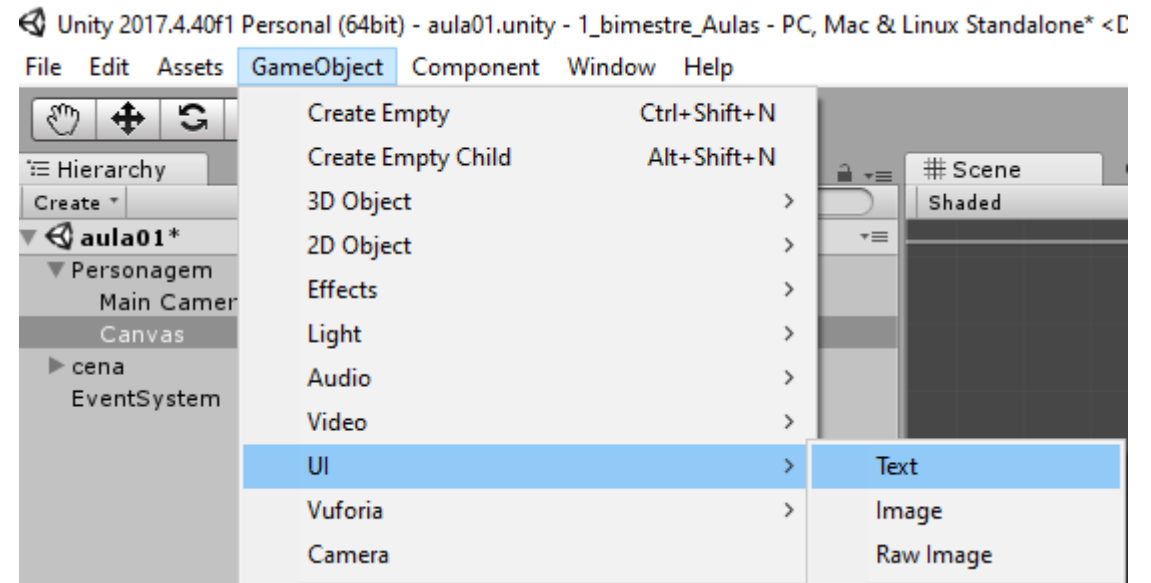


Apresentação de textos em Unity - Componente “Text”

Prof. Me. Hélio Esperidião

Adicionar um componente de texto

- GameObject >> UI >> Text



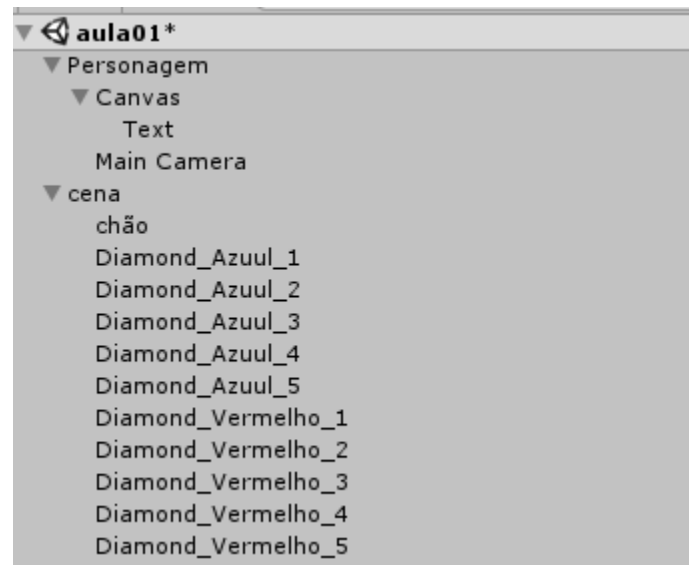
Canvas unity

- O Canvas do Unity é um componente gráfico que fornece um espaço de tela para que os elementos da interface do usuário (UI) sejam renderizados. Ele é usado para criar a interface do usuário do jogo, incluindo botões, textos, imagens e outros elementos interativos.



Mova o canvas para dentro do personagem

- Ao mover o canvas para dentro do personagem é possível que o canvas se movimente junto com o personagem.

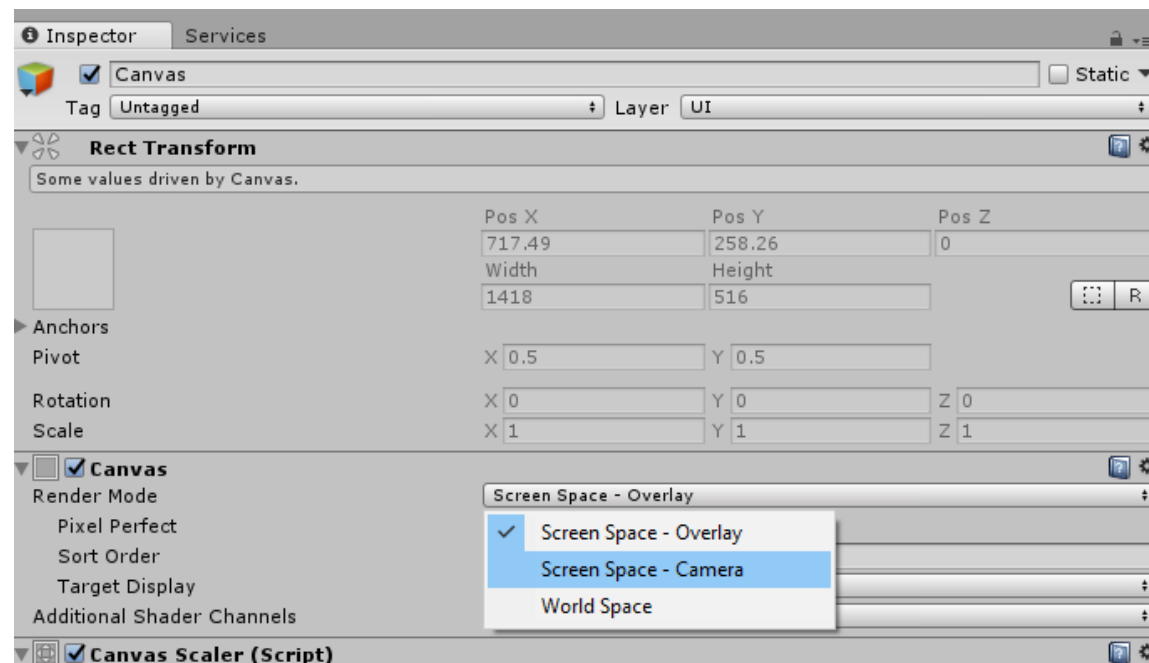
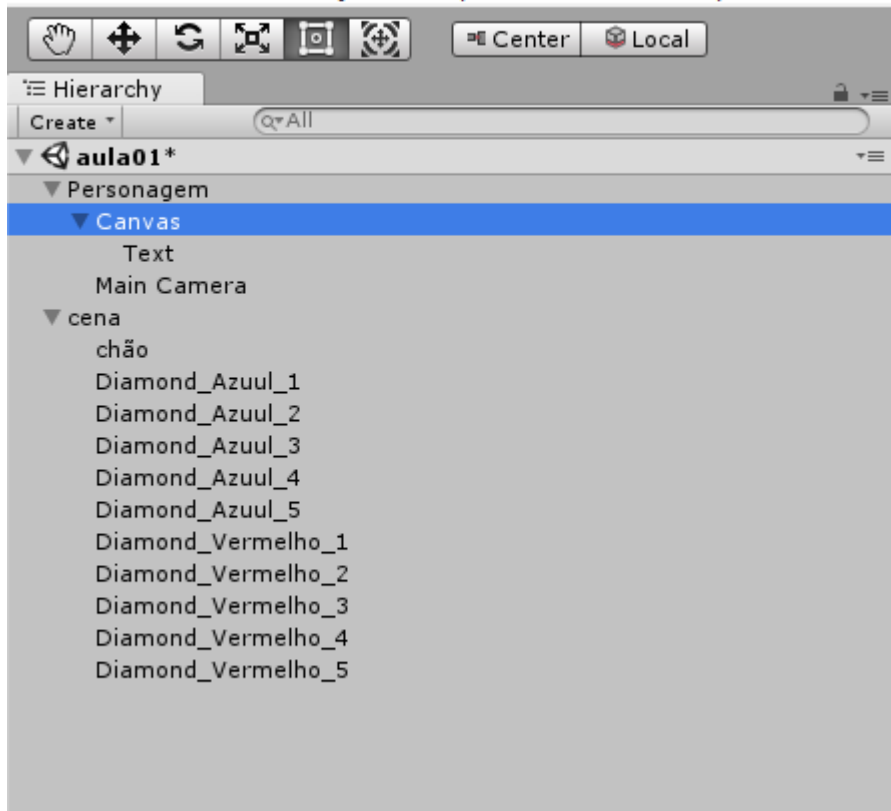


Altere a renderização de câmera

- Clique no canvas posicionado na hierarquia do projeto, no inspector mude a propriedade render mode para “Screen Space - Camera”

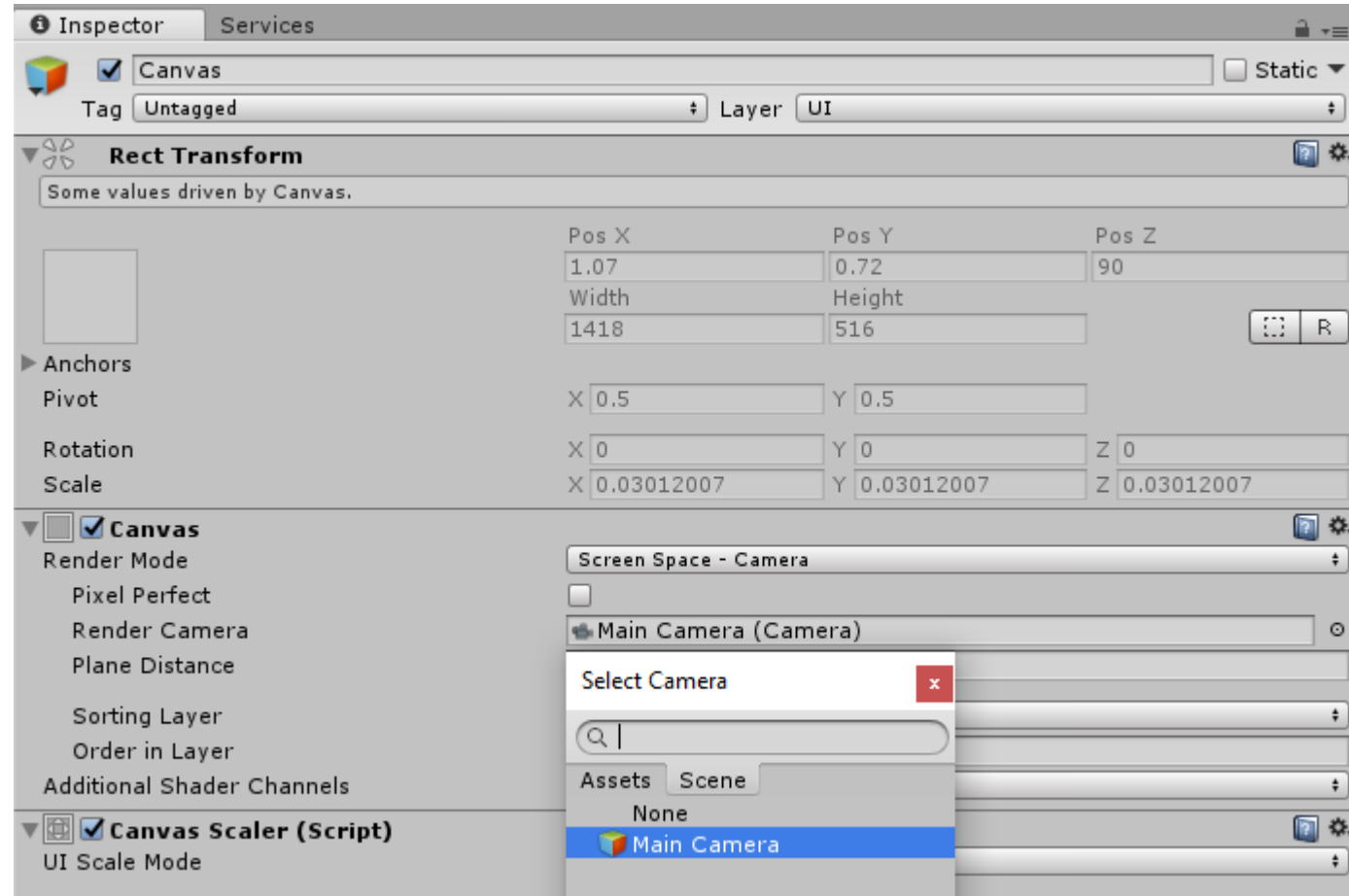
Unity 2017.4.40f1 Personal (64bit) - aula01.unity - 1_bimestre_Aulas - PC, Mac &

File Edit Assets GameObject Component Window Help



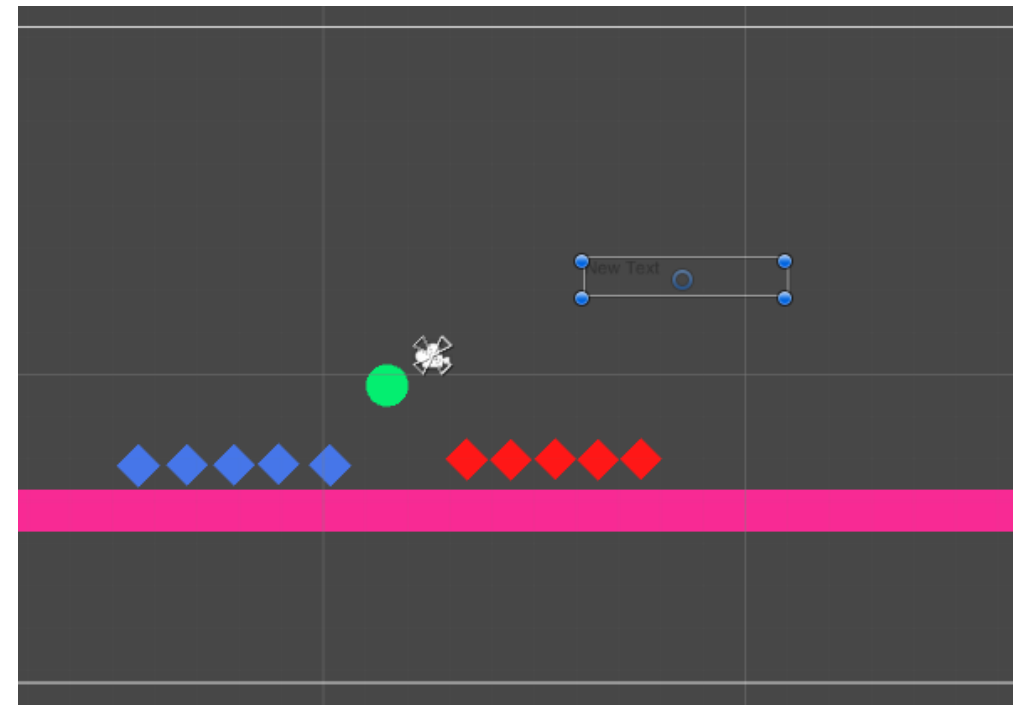
Render Camera

- Clique duas vezes na propriedade “Render Camera” e selecione “Main Camera” com dois cliques.



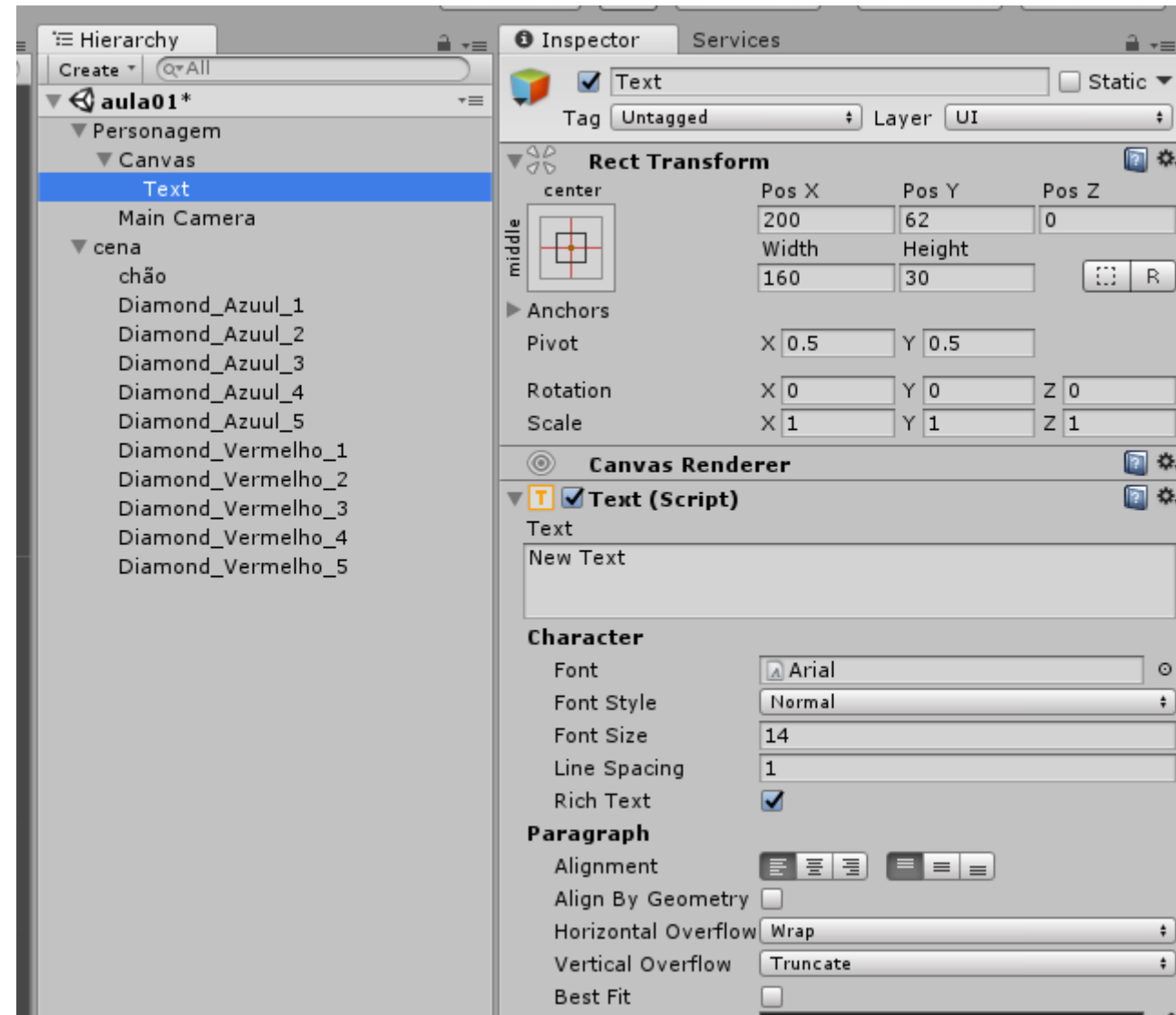
Configure os componentes

- **Posicione livremente o componente de texto no cenário.**
- Ao tocar em “t1” o personagem ganha **1** ponto.
Ao tocar em “t2” o personagem ganha **2** pontos.
- Configurações:
 - Diamante azul
 - Colisor
 - Tag: “tipo1”
 - Diamante vermelho
 - Colisor
 - Tag: “tipo2”



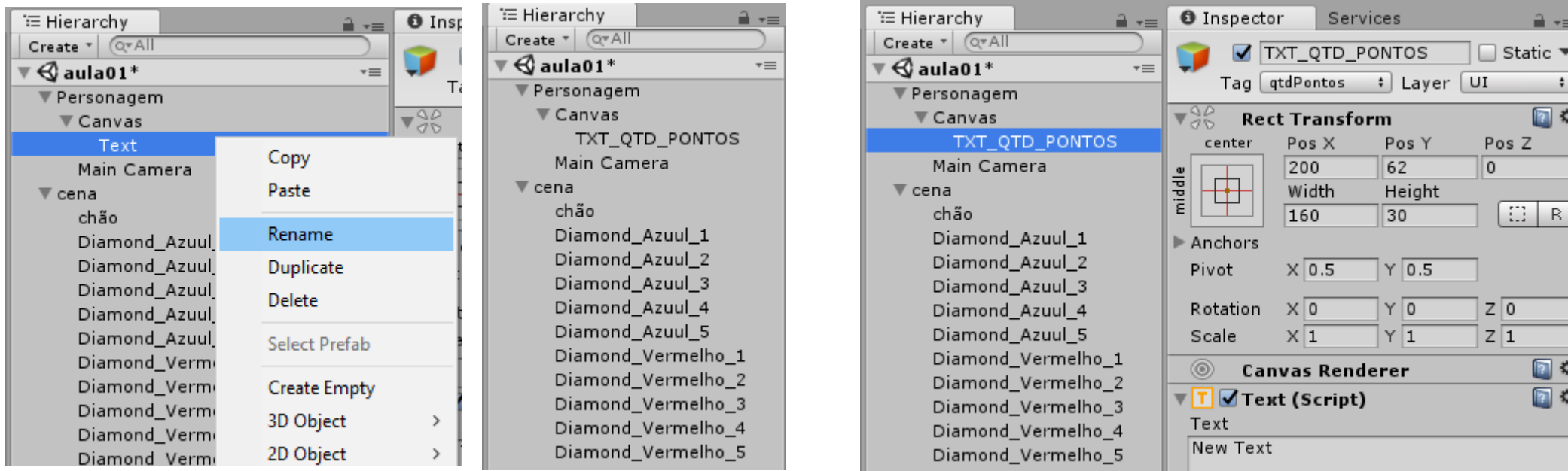
Propriedades de texto

- Selecione o componente de texto e altere propriedades de texto como cor, negrito e itálico, alinhamento e etc.



Altere o nome do componente

- Para identificar o componente de texto na programação é necessário modificar o seu nome. O exemplo modifica o nome de “Text” para “TXT_QTD_PONTOS”.
- Acrescente a tag “qtdPontos” no componente “TXT_QTD_PONTOS”



Codificação

- Importe a biblioteca:

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
using UnityEngine.UI;
```

- Biblioteca utilizada para poder trabalhar com o UI >> Text

Defina as propriedades

```
float Vx; //velocidade em x
```

```
float Vy; //velocidade em y
```

```
float VelocidadeHorizontal;
```

```
float VelocidadePular;
```

```
float DirecaoHorizontal;
```

```
Rigidbody2D CorpoRigido;
```

```
CircleCollider2D Colisor;
```

```
Text UiTextPontos; // Componente UI do tipo TEXT
```

```
int QtdPontos; // Armazena a quantidade de pontos
```

Método Start()

```
void Start () {  
    QtdPontos = 0;  
    VelocidadeHorizontal = 5;  
    VelocidadePular = 5;  
    CorpoRigido = GetComponent<Rigidbody2D> ();  
    Colisor = GetComponent<CircleCollider2D> ();  
    //inicializa o componente UI de Texto  
    //encontra o componente com a tag "qtdPontos" e recupera o Objeto Text.  
    UiTextPontos = GameObject.FindGameObjectWithTag ("qtdPontos").GetComponent<Text> ();  
    CorpoRigido.freezeRotation = true;  
    CorpoRigido.gravityScale = 1;  
}
```

OnCollisionEnter2D()

```
void OnCollisionEnter2D(Collision2D objetoTocado){  
    string tagTocada = objetoTocado.gameObject.tag;  
    if (tagTocada == "tipo1" ) {  
        QtdPontos=QtdPontos+1; //soma 1 em QtdPontos  
        string textoPontos= QtdPontos.ToString();  
        UiTextPontos.text = textoPontos;  
        Destroy (objetoTocado.gameObject);//destrói tipo1  
    }  
    if (tagTocada == "tipo2" ) {  
        QtdPontos=QtdPontos+2; soma 2 em QtdPontos  
        string textoPontos= QtdPontos.ToString();  
        UiTextPontos.text = textoPontos;  
        Destroy (objetoTocado.gameObject); //destrói tipo2  
    }  
}
```

Transforma um número em texto
Pois: `UiTextPontos.text`
só é capaz de receber
variáveis
do tipo texto