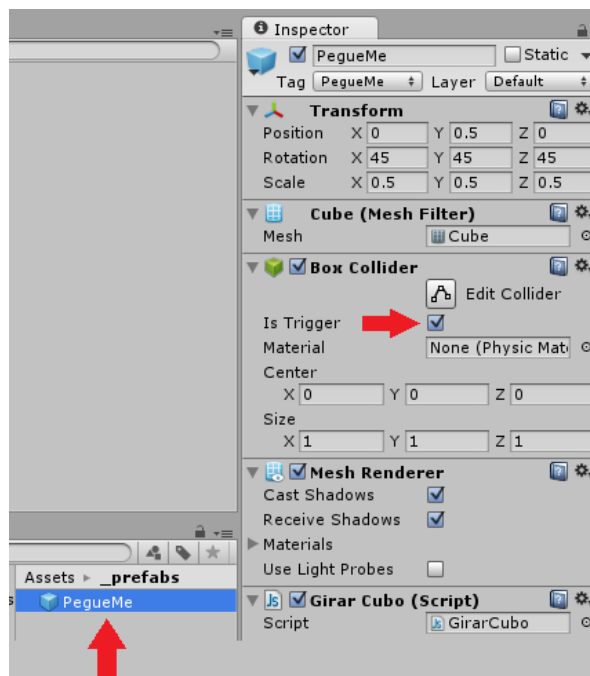


LABORATÓRIO: CRIANDO O JOGO **ROLL A BALL**

Parte-2

1. Agora vamos inserir no jogo o comportamento de destruir os cubos quando a esfera tocar neles e, assim, marcar pontos para o jogador. Para fazer isto, precisamos de uma forma de detectar quando ocorre uma **colisão** entre o objeto esfera com um dos objetos *PegueMe*. Então precisamos inserir 3 comportamentos (*behaviors*) ao jogo, sendo eles:
 - a. Detectar a colisão entre os objetos esfera e *PegueMe*, disparando um evento;
 - b. Remover do jogo o objeto *PegueMe* com o qual a esfera colidiu;
 - c. Marcar pontos para o jogador.
2. Para detectar a colisão entre a esfera e os objetos *PegueMe*, iremos marcar a propriedade **"Is Trigger"** do componente **"Box Collider"** do prefab *PegueMe* (ver figura). Esta alteração será refletida automaticamente em todos os objetos *PegueMe* criados na cena. Marcando esta propriedade, irá fazer com que seja disparado o evento **OnTriggerEnter** no objeto que colidir com o objeto *PegueMe*. No nosso caso, será disparado um evento **OnTriggerEnter** para o objeto esfera.



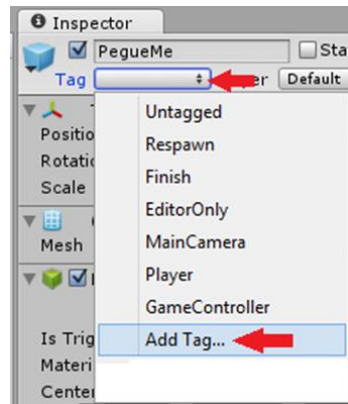
3. Agora iremos programar o evento **OnTriggerEnter** da esfera para que o objeto *PegueMe* seja removido da cena. Abra o script *ControlarJogador* (que está associado à esfera) e insira nele a função listada a seguir. Esta função possui um argumento chamado **other** (do tipo *Collider*) e através dele podemos ter acesso ao objeto com o qual a esfera colidiu.

```
//essa funcao eh chamada quando um Collider de outro
//objeto colide com o objeto associado ao script
function OnTriggerEnter (other: Collider) {
    //verifica se a Tag do outro objeto eh "PegueMe"
    if (other.gameObject.CompareTag("PegueMe")) {
        //desativa o outro objeto, tornando-o invisivel
        other.gameObject.SetActive(false);
    }
}
```

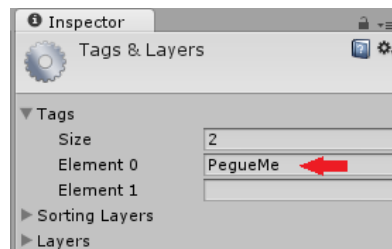
4. Para que o código criado no passo anterior funcione corretamente, precisamos de uma forma de identificar se a esfera colidiu com um objeto do tipo *PegueMe*, pois ela poderá colidir com outros tipos de objetos também, como os "muros" que criamos em volta do plano. Na primeira linha de código da função está sendo feita uma comparação (*if*) para verificar se o outro objeto (*other*) é um objeto *PegueMe* e, caso seja, desativamos ele para que fique invisível na cena. Aqui usamos a

propriedade **Tag** do objeto para fazer a verificação, através do método **CompareTag()**. A **Tag** pode ser utilizada para identificar um objeto (ou grupo de objetos) dentro do nosso jogo. Iremos então definir a **Tag** do nosso prefab *PegueMe* para o texto “PegueMe” e, com isso, todos os objetos deste tipo na cena também receberão essa mesma **Tag**.

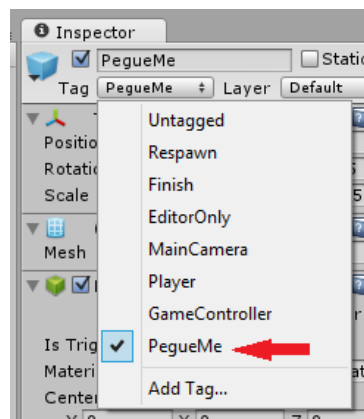
- a. Selecione o prefab *PegueMe* e clique na sua propriedade **Tag** (ela fica logo abaixo do nome do objeto). Acesse a opção **Add Tag** (ver figura).



- b. Insira o texto “PegueMe” dentro do “Element 0” de Tags (ver figura).



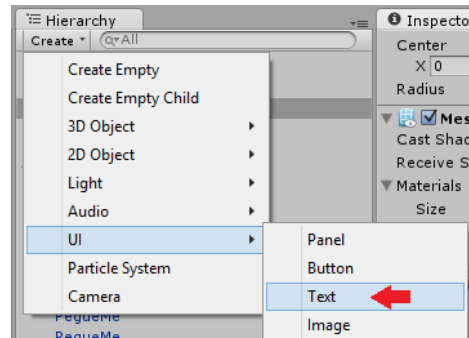
- c. Selecione novamente o prefab *PegueMe* na janela *Project*, abra a sua propriedade **Tag** e verifique que agora a opção “PegueMe” está disponível para seleção. Marque-a conforme a figura.



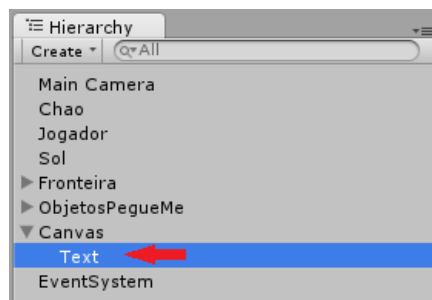
- d. Pronto! Com isso já podemos identificar todos os objetos *PegueMe* do jogo através da sua **Tag**. E o código dentro da função *OnTriggerEnter()* irá funcionar corretamente.

5. Execute e teste seu jogo.

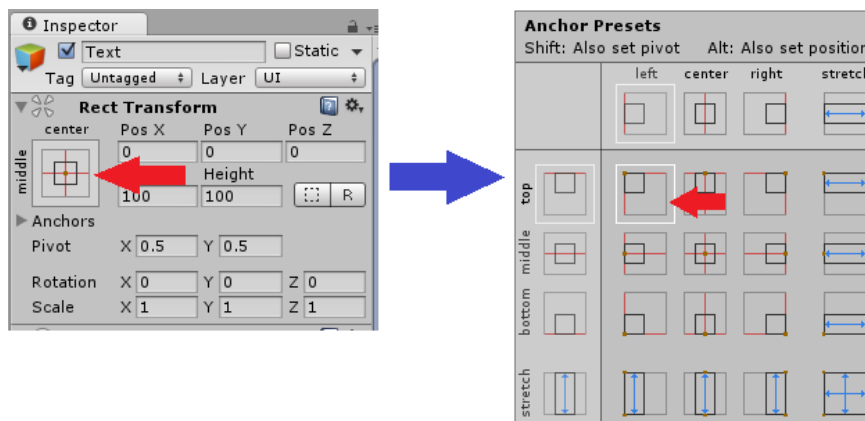
6. Vamos agora inserir o mostrador de pontos do jogador. Insira um objeto *Text* a partir da opção *Create* da janela *Hierarchy* (ver figura).



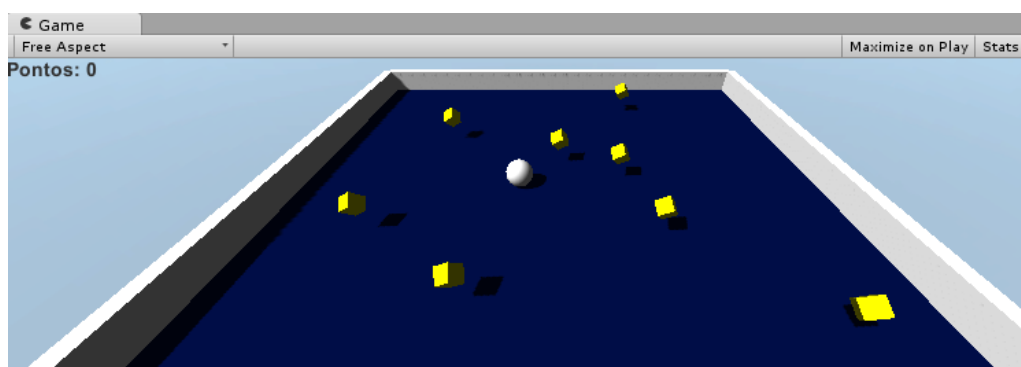
7. Perceba que ele será criado dentro de um *game-object* chamado **Canvas** (ver figura). O *Canvas* é um objeto reservado para criarmos os objetos de UI (*User Interface*) do jogo. Portanto, todos os objetos de UI que precisarmos deverá ser criado dentro de um objeto *Canvas*. Observe que também foi criado outro objeto, do tipo *EventSystem*, mas neste projeto não vamos nos preocupar com ele.



8. Selecione na janela *Hierarchy* o objeto *Text* criado no passo anterior. Vamos então posicioná-lo no canto superior esquerdo da tela. Clique no botão de posicionamento do componente *Rect Transform* e em seguida, pressionando a tecla *Alt*, selecione o posicionamento *Top* e *Left* (ver figura). Tecle *Enter* para confirmar as mudanças. Com isso o texto já ficará posicionado onde queremos.



9. Altere o texto do objeto *Text* para "Pontos: 0" e deixe-o em negrito. O jogo deverá ficar parecido com a figura a seguir:



10. Agora abra o script *ControlarJogador* e faça as alterações destacadas em vermelho na figura a seguir. Aqui definimos duas novas variáveis, sendo uma para acumular os pontos marcados (*pontos*) e outra, do tipo *Text*, que será a nossa referência ao objeto que foi inserido na cena para exibir os pontos (*mostradorPontos*). Criamos também uma função chamada *AtualizarPontos()* que será responsável por alterar o texto do objeto *mostradorPontos*. Na função *Start()* zeramos os pontos do jogador e na função *OnTriggerEnter* acrescentamos pontos cada vez que ele colidir com um objeto *PegueMe*.

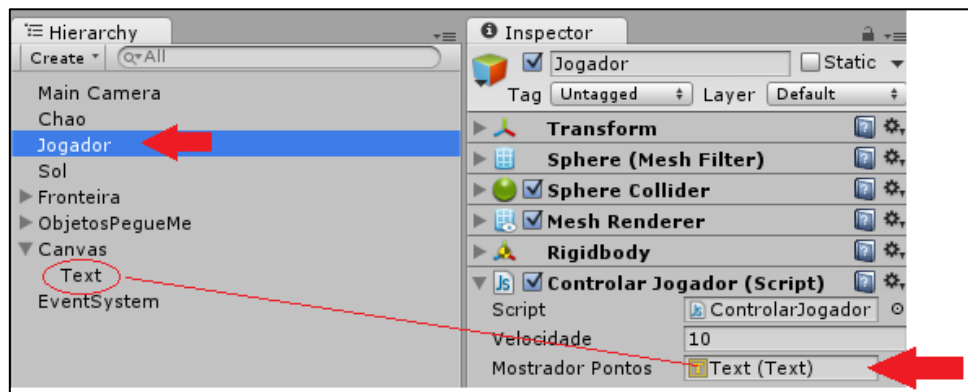
```
private var pontos: int;
public var mostradorPontos: UnityEngine.UI.Text;

function Start () {
    //obtem o componente Rigidbody do gameObject
    rb = gameObject.GetComponent.<Rigidbody>();
    //zera os pontos e atualiza na tela
    pontos = 0;
    AtualizarPontos();
}

//essa funcao eh chamada quando um Collider de outro
//objeto colide com o objeto associado ao script
function OnTriggerEnter (other: Collider) {
    //verifica se a Tag do outro objeto eh "PegueMe"
    if (other.gameObject.CompareTag("PegueMe") ) {
        //desativa o outro objeto, tornando-o invisivel
        other.gameObject.SetActive(false);
        //marca mais 1 ponto para o jogador e atualiza na tela
        pontos++;
        AtualizarPontos();
    }
}

//atualiza os pontos na tela do jogo
function AtualizarPontos(){
    mostradorPontos.text = "Pontos: " + pontos;
}
```

11. Agora volte para a IDE do Unity, selecione o objeto *Jogador* e defina na propriedade “*Mostrador Pontos*” o objeto *Text*.



12. Execute e teste o jogo.

13. Agora insira um novo objeto *Text* dentro de *Canvas*, renomeie para *WinText*, e altere seu texto para “**Você Venceu!!!**”. Na propriedade *Paragraph*, mude os alinhamentos para “*Center Align*” e “*Middle Align*”. Mude também as propriedades *Horizontal Overflow* e *Vertical Overflow* para “*Overflow*”. Altere o tamanho da fonte para 39 e a cor para azul claro. A cena deverá ficar semelhante à figura a seguir.



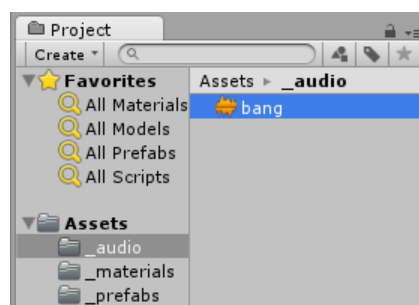
14. Abra o script *ControlarJogador* e faça as alterações a seguir.

```
private var pontos: int;
public var mostradorPontos: UnityEngine.UI.Text;
public var textoVoceVenceu: UnityEngine.UI.Text;

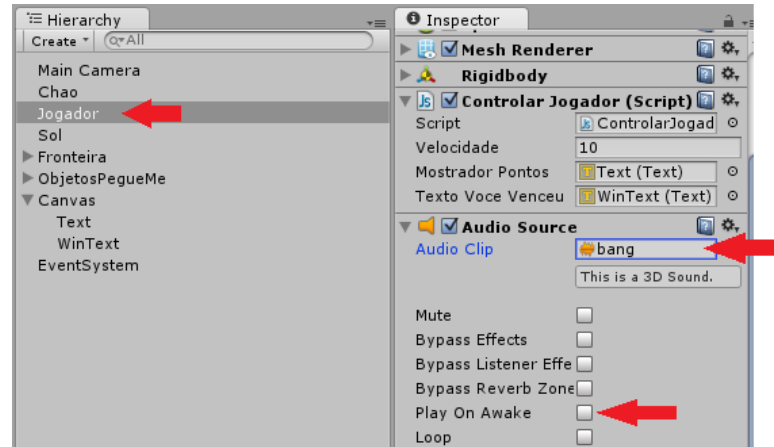
function Start () {
    //obtem o componente Rigidbody do gameObject
    rb = gameObject.GetComponent.<Rigidbody>();
    //zera os pontos e atualiza na tela
    pontos = 0;
    AtualizarPontos();
    //limpa a mensagem "Voce Venceu"
    textoVoceVenceu.text = "";
}

//atualiza os pontos na tela do jogo
function AtualizarPontos(){
    mostradorPontos.text = "Pontos: " + pontos;
    if (pontos == 8) {
        textoVoceVenceu.text = "Voce Venceu!!!";
    }
}
```

15. Agora volte para a IDE do Unity, selecione o objeto *Jogador* e defina na propriedade “*Texto Voce Venceu*” o objeto *WinText*.
16. Execute e teste o jogo.
17. Vamos fazer com que seja emitido um som quando a esfera tocar nos objetos *PegueMe*. Baixe o arquivo de áudio disponibilizado na página do professor (*bang.wav*) e adicione-o ao projeto através do menu “*Assets->Import New Asset*”. Na janela *Project*, crie uma pasta chamada “*_audio*” e mova o arquivo de áudio para dentro dela (ver figura).



18. Selecione o objeto *Jogador* na janela *Hierarchy* e adicione um componente *AudioSource* para ele, através da opção “*Add Component->Audio->Audio Source*”. Defina a propriedade *Audio Clip* para o nosso arquivo de áudio e desmarque a opção *Play on Awake* (ver figura).



19. Abra o script *ControlarJogador* e faça as alterações a seguir. Aqui obtemos uma referência ao objeto *AudioSource* da esfera e tocamos o áudio quando for detectada uma colisão com um dos objetos *PegueMe* da cena.

```
private var audioSrc: UnityEngine.AudioSource;

function Start () {
    //obtem o componente Rigidbody do gameObject
    rb = gameObject.GetComponent.<Rigidbody>();
    //zera os pontos e atualiza na tela
    pontos = 0;
    AtualizarPontos();
    //limpa a mensagem "Voce Venceu"
    textoVoceVenceu.text = "";
    //obtem uma referencia ao objeto AudioSource
    audioSrc = gameObject.GetComponent.<AudioSource>();
}

//essa funcao eh chamada quando um Collider de outro
//objeto colide com o objeto associado ao script
function OnTriggerEnter (other: Collider) {
    //verifica se a Tag do outro objeto eh "PegueMe"
    if (other.gameObject.CompareTag("PegueMe") ) {
        //desativa o outro objeto, tornando-o invisivel
        other.gameObject.SetActive(false);
        //marca mais 1 ponto para o jogador e atualiza na tela
        pontos++;
        AtualizarPontos();
        //toca o som de colisao
        audioSrc.Play();
    }
}
```

20. Execute e teste o jogo.