

Criando Um Protótipo do Snake

Tiago Machado

tlam@cin.ufpe.br

Movimento

- Para o movimento usaremos a função abaixo
- O script com a função deve ser anexado ao objeto **Ponteiro**

```
function Movimento () {  
  
    transform.Translate(Vector3(0, 0, velocidade));  
  
    if (Input.GetKeyDown(KeyCode.LeftArrow)) {  
        transform.Rotate(0, 90, 0);  
    }  
    if (Input.GetKeyDown(KeyCode.RightArrow)) {  
        transform.Rotate(0, -90, 0);  
    }  
  
}
```

- Note que o personagem sempre se move no seu eixo Z positivo e as setas do teclado (esquerda e direita) mudam a direção do movimento em 90 graus

Criando Objetos

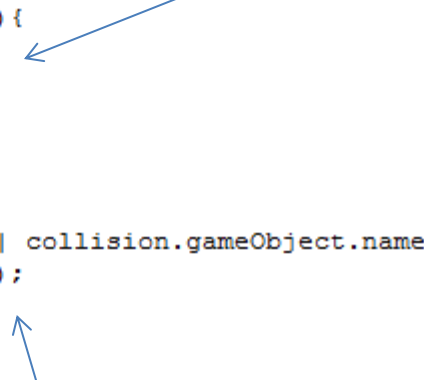
- Para criar os objetos, vamos usar o script `CriarObjetos`
- O mesmo utiliza corrotina para criar os objetos em intervalos de tempo definidos
- A função `Start` inicia a corrotina com a chamada `StartCorroutine("InstanciarObjeto");`
- A função iniciada pela corrotina só executará após o tempo passado como parametro em `yield WaitForSeconds(tempo);`
- Após isso a função cria o objeto em posição aleatória, mas dentro da arena
- Por fim retorna a função `Start` para reiniciar através da corrotina o mesmo processo de criação novamente.

Colisão

Temos de considerar três tipos de colisão:

- Com objetos
 - Esses serão destruídos e o snake ganhará um novo segmento

```
function OnCollisionEnter (collision:Collision) {  
    if (collision.gameObject.tag == "maca") {  
        novoSegmento ();  
        Destroy (collision.gameObject);  
    }  
  
    if (collision.gameObject.tag == "parede" || collision.gameObject.name == "seg") {  
        Application.LoadLevel ("SnakeGameOver");  
    }  
}
```

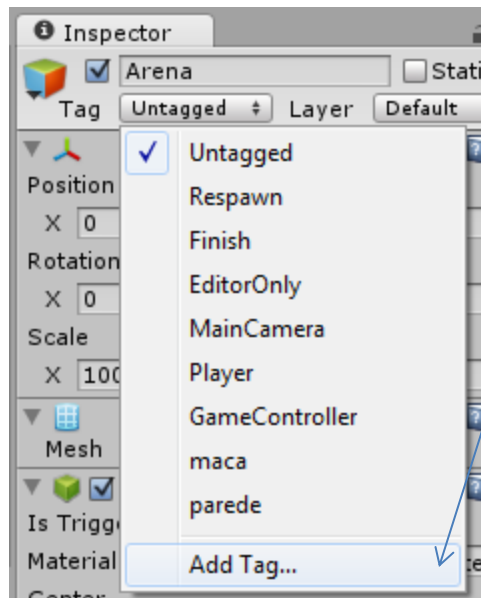


- Com Paredes e Segmentos
 - Em ambos os casos o jogador perde e mudamos para uma cena na qual imprimimos **Game Over** na tela.

Colisão

As colisões consideram as **tags** dos objetos colididos.

Para criar **tags** clique em um **GameObject** e adicione sua tag clicando no botão do menu abaixo



Tags são muito úteis quando você precisa criar especificações nos seus **Game Objects**, por exemplo: especificar que alguns são **moedas**, outros **inimigos**, etc.

Criando Novo Segmento

Quando o código inicia, indicamos que o último segmento é o **GameObject** que representa o ponteiro do Snake

```
function Start () {  
    renderer.material.color = Color.blue;  
    seg = gameObject;  
}
```

Quando um novo segmento é criado, devemos fazer com que o mesmo siga o último segmento, para isso usamos o script **Smooth Follow** (que é do pacote de standard assets do Unity).

```
function novoSegmento() {  
    var segmento:GameObject = Instantiate(novo, Vector3(30, 0, 200), Quaternion.identity);  
    segmento.renderer.material.color = Color.blue;  
    segmento.name = "seg";  
    segmento.GetComponent(SmoothFollow).target = seg.transform;  
    seg = segmento;  
}
```

Devemos atualizar o novo segmento como o último segmento

Esses são apenas os valores iniciais onde o novo segmento aparecerá – mas na prática não tem tanta influencia já que o SmoothFollow o fará ficar sempre atrás do segmento anterior

Execute o exemplo

Todos os scripts e assets do protótipo estão disponíveis no material desta aula, descompacte o pacote dentro do Unity, execute e confira.

Referências