

02

Rotação do jogador 2

Transcrição

Anteriormente:

- criamos o ponto de impacto do raio, a partir da posição de "Jogador", e o salvamos na variável `posicaoMiraJogador`;
- cancelamos o eixo Y para igualar a altura de ponto de impacto e "Jogador".

Agora, utilizaremos a mesma variável (`posicaoMiraJogador`) para fazer a rotação de "Jogador".

Começaremos criando um `Quaternion` para validar a rotação. O nomearemos como `novaRotacao` e atribuiremos (`=`) como valor `Quaternion.LookRotation`, que calculará uma rotação a partir da posição de mira de "Jogador".

Para inserir a rotação na personagem, utilizaremos `GetComponent<> . O if()`, em `FixedUpdate` ficará da seguinte forma:

```
if(Physics.Raycast(raio, out impacto, 100))
{
    Vector3 posicaoMiraJogador = impacto.point - transform.position;

    posicaoMiraJogador.y = transform.position.y;

    Quaternion novaRotacao = Quaternion.LookRotation(posicaoMiraJogador);

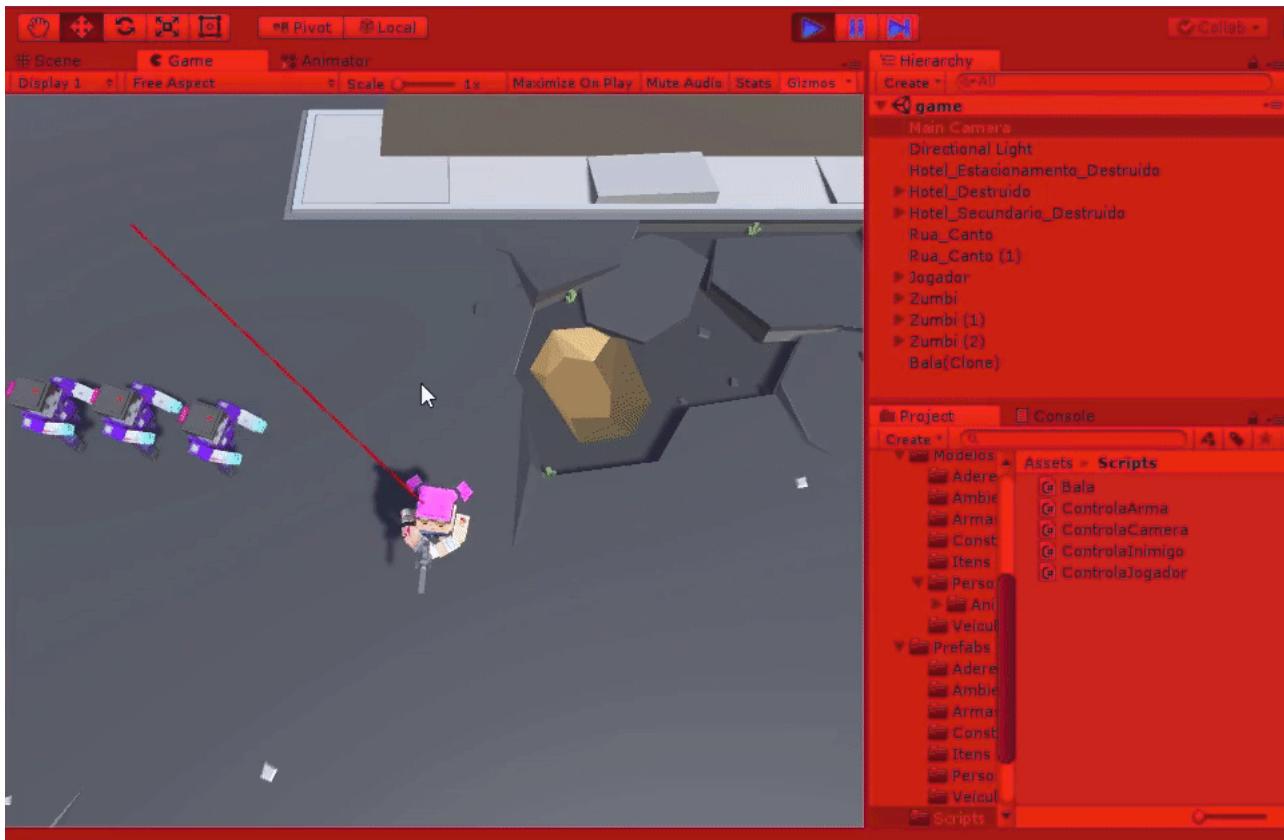
    GetComponent<Rigidbody>().MoveRotation(novaRotacao);

}
```

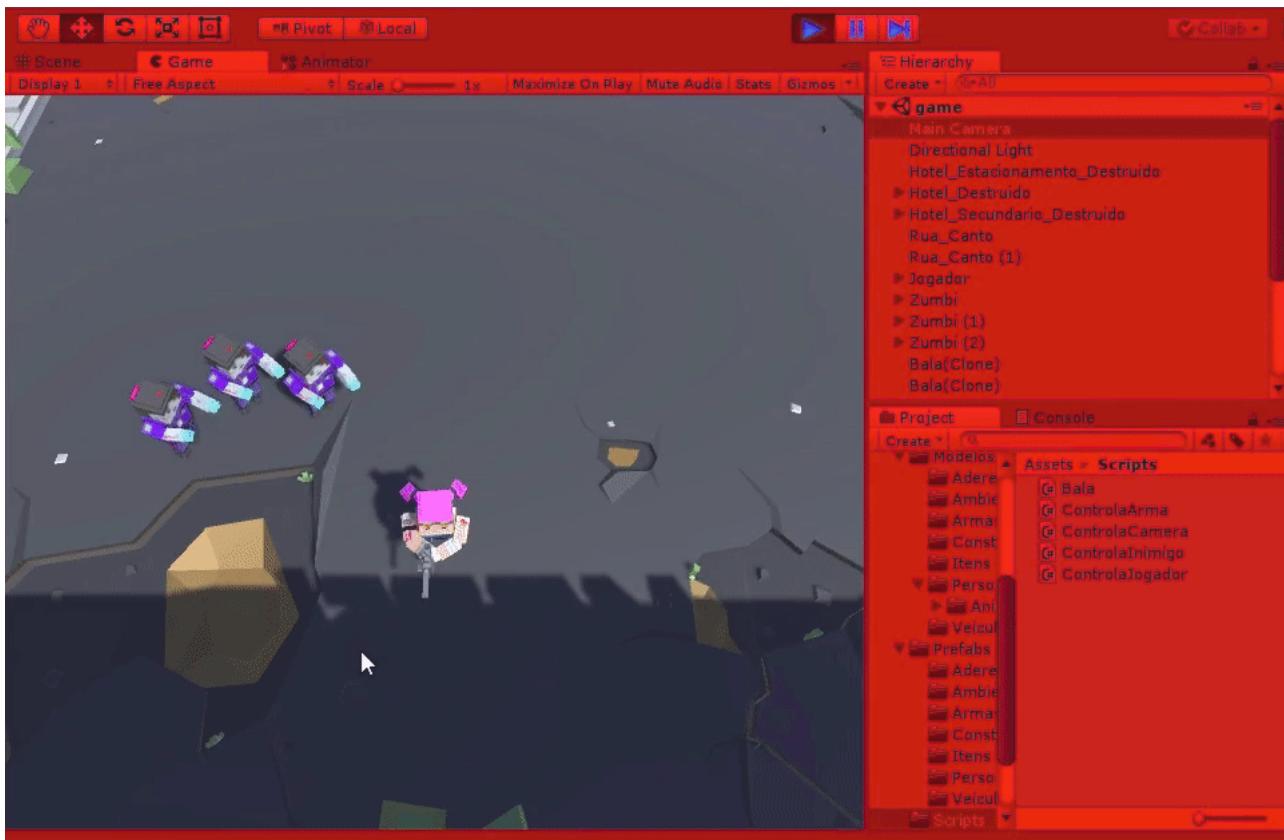
Assim:

- criamos um raio e um impacto e, em `Raycast`, juntamos os dois;
- a partir da posição do impacto, com base na posição de "Jogador", geramos uma posição de mira;
- cancelamos o eixo Y, para que ponto de impacto e "Jogador" fiquem na mesma altura;
- geramos uma variável `novaRotacao` equivalente à rotação de acordo com a posição, ou seja, calculamos a rotação a partir da posição de mira de "Jogador", e a lançamos em `MoveRotation()`.

Salvaremos e minimizaremos o *script*. De volta à Unity, arrastaremos a janela de "Game" para cima e ativaremos "Play". Veremos "Jogador" e o raio vermelho acompanhando o ponteiro do mouse, porque estamos com a parte de *gizmo* marcada, que mostra o que fizemos no `Debug`.



Desmarcaremos e veremos que "Jogador", está rotacionando. Mas, notem que a rotação está estranha.



Em um momento a personagem está virada para um lado e, do nada, vira para o outro. Isso acontece porque o raio está batendo nas estruturas do cenário e, ele não entende o que acontece quando bate em algo. O ideal é que ele bata somente no chão. Para corrigir, limitaremos o impacto do raio ao chão.

Acima de `Update`, declararemos uma variável pública do tipo `LayerMask`, uma camada que geraremos para mascarar tudo, menos o chão. A nomearemos como `MascaraChao` e a colocaremos em `Raycast()`. O código de

ControlaJogador.cs ficará da seguinte forma:

```
public class ControlaJogador : MonoBehaviour {

    public float Velocidade = 10;
    Vector3 direcao;
    public LayerMask MascaraChao;

    // Update is called once per frame
    void Update () {

        float eixoX = Input.GetAxis("Horizontal");
        float eixoZ = Input.GetAxis("Vertical");

        direcao = new Vector3(eixoX, 0, eixoZ);

        if(direcao != Vector3.zero)
        {
            GetComponent<Animator>().SetBool("Movendo", true);
        }
        else
        {
            GetComponent<Animator>().SetBool("Movendo", false);
        }

    }

    void FixedUpdate()
    {
        GetComponent<Rigidbody>().MovePosition
            (GetComponent<Rigidbody>().position +
            (direcao * Velocidade * Time.deltaTime));

        Ray raio = Camera.main.ScreenPointToRay(Input.mousePosition);
        Debug.DrawRay(raio.origin, raio.direction * 100, Color.red);

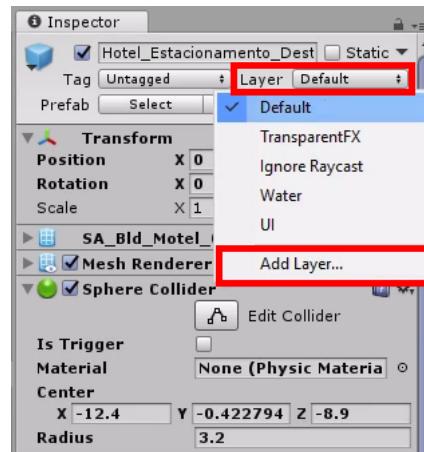
        RaycastHit impacto;

        if(Physics.Raycast(raio, out impacto, 100, MascaraChao))
        {
            Vector3 posicaoMiraJogador = impacto.point - transform.position;

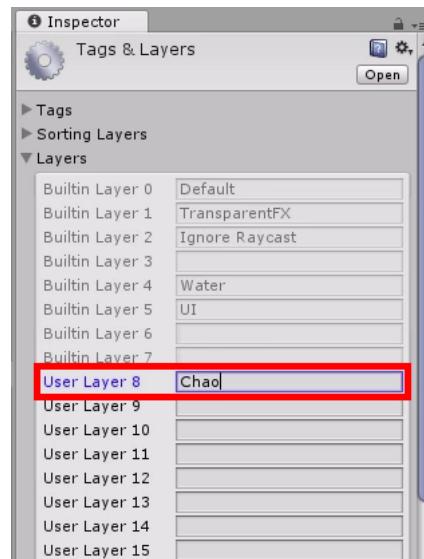
            posicaoMiraJogador.y = transform.position.y;

            Quaternion novaRotacao = Quaternion.LookRotation(posicaoMiraJogador);
            GetComponent<Rigidbody>().MoveRotation(novaRotacao);
        }
    }
}
```

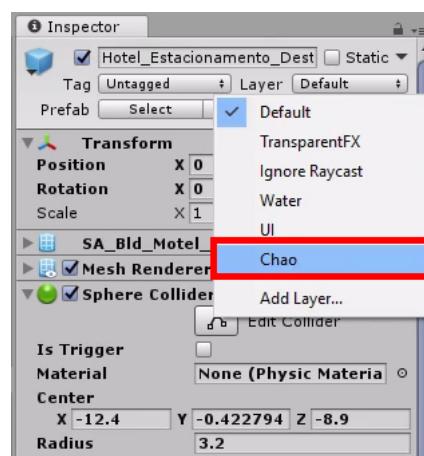
Assim, o raio baterá somente nos objetos que estão nas camadas com `MascaraChao`. Para usá-la, salvaremos e minimizaremos o `script`; voltaremos à Unity e, em "Inspector" de "Jogador", faremos "Mascara Chao" pegar somente o chão. Clicaremos nele em "Scene" e, no canto superior direito de "Inspector", definiremos uma camada para ele em "Layer" selecionando "Default > Add Layer".



Em "Layer 8", digitaremos "Chao".



Clicaremos novamente no chão, em "Scene", e acrescentaremos a camada que criamos, clicando em "Layers" para alterar o padrão ("Default") para "Chao", camada que acabamos de criar. Assim, colocamos os objetos em um lugar separado do chão.



Usaremos a camada em função da "Mascara Chao". Por enquanto, o raio não bate em nada porque no "Inspector" de "Jogador", está definido como "Nothing". Clicaremos em "Nothing" e mudaremos para "Chao", assim conseguimos separar o chão do resto do cenário para que o raio bata somente nele.



Ao ativarmos "Play" novamente, veremos que "Jogador" está rotacionando perfeitamente.

