



07

Diminuindo o alcance da Torre

Para dificultar um pouco mais o jogo e fazer com que o jogador tenha que pensar onde colocar as torres futuramente, vamos fazer com que a torre tenha alcance limitado e que ela tenha que escolher um alvo dentro desse alcance quando precisar disparar os mísseis.

- Adicione um novo atributo no script `Torre` para indicar o alcance da torre:

```
public class Torre : MonoBehaviour
{
    [SerializeField] private float raioDeAlcance;
}
```

- Selecione a Torre em Hierarchy e no Inspector -> Torre coloque Raio de Alcance = 2 .
- Agora, nossa torre vai ter que ser capaz de escolher entre múltiplos alvos. Para organizar nosso script, vamos criar um novo método no script `Torre` que ficará responsável por escolher um alvo:

```
public class Torre : MonoBehaviour
{
    private Inimigo EscolheAlvo()
    {
        GameObject[] inimigos = GameObject.FindGameObjectsWithTag("Inimigo");
        foreach (GameObject inimigo in inimigos)
        {
            //Como saberemos em qual alvo atirar?
        }
        return null;
    }
}
```

- Novamente, para manter nosso script organizado vamos criar mais um método que ficará responsável por dizer se um inimigo se encontra dentro do alcance da torre:

```
public class Torre : MonoBehaviour
{
    private bool EstaNoRaioDeAlcance(GameObject inimigo)
    {
        Vector3 posicaoDoInimigoNoPlano =
            Vector3.ProjectOnPlane(inimigo.transform.position, Vector3.up);
        Vector3 posicaoDaTorreNoPlano =
            Vector3.ProjectOnPlane(this.transform.position, Vector3.up);

        float distanciaParaInimigo =
            Vector3.Distance (posicaoDaTorreNoPlano, posicaoDoInimigoNoPlano);

        return distanciaParaInimigo <= raioDeAlcance;
    }
}
```

- Pronto! Agora sabemos em qual alvo devemos atirar: basta verificar se ele está no raio de alcance da nossa torre:

```
public class Torre : MonoBehaviour
{
    //...

    private Inimigo EscolheAlvo()
    {
        GameObject[] inimigos = GameObject.FindGameObjectsWithTag("Inimigo");
        foreach (GameObject inimigo in inimigos)
        {
            if (EstaNoRaioDeAlcance(inimigo))
            {
                return inimigo.GetComponent<Inimigo>();
            }
        }
        return null;
    }
}
```

- Para finalizar, precisamos fazer com que nossa torre utilize o método `EscolheAlvo`. Vamos então alterar o método `Update`:

```
public class Torre : MonoBehaviour
{
    void Update ()
    {
        Inimigo alvo = EscolheAlvo ();
        if (alvo != null)
        {
            //É só passar o alvo para o método atira...
            Atira (alvo);
        }
    }
}
```

Neste momento, estamos com um problema de compilação no nosso código. O que aconteceu?

Responda

INSERIR CÓDIGO	FORMATAÇÃO

