

Game Over quando acabarem as vidas (parte 2)

Podemos criar um método `JogoAcabou` para se chamado sempre que nosso jogo chegar ao fim, mas o que significa isso?

```
public class Jogo : MonoBehaviour
{
    void Update ()
    {
        if (JogoAcabou ())
        {
            gameOver.SetActive(true);
        } else
        {
            if (ClicouComBotaoPrimario ())
            {
                ConstroiTorre ();
            }
        }
    }

    private bool JogoAcabou ()
    {
        //Quando o jogo acaba??
    }
}
```

Nosso jogo acaba quando o jogador morre, ou seja, quando sua vida chegar a zero. Então, podemos criar um método que `EstaVivo` na classe `Jogador` :

```
public class Jogador : MonoBehaviour
{
    public bool EstaVivo ()
    {
        return vida > 0;
    }
}
```

E chamá-lo na nossa classe `Jogo` :

```
public class Jogo : MonoBehaviour
{
    [SerializeField] private Jogador jogador;

    private bool JogoAcabou ()
    {
        return !jogador.EstaVivo ();
    }
}
```

Como nossa classe `Jogo` agora recebe um `Jogador`, precisamos de um `GameObject`. Então, pelo Unity vamos selecionar nosso `Hierarchy/Jogo` e arrastar `Hierarchy/DadosDoJogador` para `Inspector -> Jogo -> Jogador`.

Agora, vamos fazer nosso `Jogador` perder vida somente se ainda estiver vivo:

```
public class Jogador : MonoBehaviour
{
    public void PerdeVida ()
    {
        if (EstaVivo ())
        {
            vida --;
        }
    }
}
```

Clique em *Play* e nos diga em que momento aparece a mensagem de Game Over. Além disso, o problema da vida negativa do jogador foi resolvido?

Responda

INSERIR CÓDIGO	FORMATAÇÃO
<div></div>	