

Validando inputs

Ao aceitarmos qualquer *GameObject* dentro das reservas, abrimos oportunidades para que alguém da equipe coloque algum objeto ali que não implementa a interface `IReservavel` e com isso podemos ter um erro lançado pelo script da reserva assim que iniciarmos o jogo e ele tentar buscar um componente que implemente essa interface dentro do objeto que lhe foi passado.

Como podemos evitar isso?

Afinal, não queremos erros inesperados acontecendo no projeto quando alguém esqueceu de como algum componente funciona. Para isso, temos o método `OnValidate` que é chamado pela Unity sempre que o valor de um atributo do inspetor é alterado. Assim, podemos fazer as verificações necessárias dentro do nosso código da reserva antes do jogo realmente iniciar.

Assim avisamos a equipe que do que ela precisa fazer para algum componente funcionar.

Dentro desse método nós podemos fazer a verificação de se o objeto passado para o atributo *prefab* desse componente foi preenchido corretamente. Para isso devemos chamar o método `GetComponent<IReservavel>` e em seguida verificar o retorno desse método.

Se tivermos um retorno nulo sabemos que o objeto colocado via inspetor da Unity não tem nenhum componente que implementa essa interface e com isso, podemos colocar uma referência nula para a propriedade *prefab* e lançarmos uma exceção que será vista no console da Unity.

Nosso código ficaria assim:

```
private void OnValidate()
{
    var reservavel = this.prefab.GetComponent<IReservavel>();
    if (reservavel == null)
    {
        this.prefab = null;
        throw new Exception("Atributo [prefab] requer um objeto que contenha a implementação da interface IReservavel");
    }
}
```

E ao colocarmos um objeto errado via inspetor veríamos a mensagem de erro no console.

