

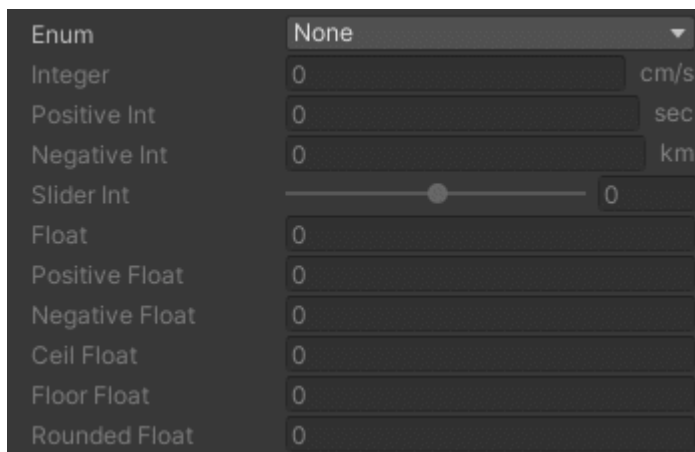
Locking properties

Descripción

Description

Allows enable or disable fields from the inspector according to a selected enumeration value or boolean property.

Reference



- **Backups** – Dropdown menu with all available backups.
- **Backup** – Create a back up of the component.
- **Restore** – Replace the current component values for the values saved in the selected back up from dropdown menu.
- **Remove** – Delete the selected backup from the dropdown menu.

Use

Select a gameobject or prefab or multiple gameobjects and prefabs with a collider or multiple colliders of the same type to to change their collider properties.

Enumeration

Each property construction method has an overload with two additional parameters to add the enumeration and the enumeration index of values that enable or disable the field.

In this example each filed method receives an “_enum” variable as an enumeration parameter and a int value as other parameter that designates wich values from “_enum” enables the field.

```
public override void OnInspectorGUI()
{
    CGFEditorUtilities.BuildEnum("Enum", "Enumeration of different values", _enum, "cm/s");
    CGFEditorUtilities.BuildInt("Integer", "All integer values", _int, "cm/s");
    CGFEditorUtilities.BuildIntPositive("Positive Int", "Only positive integer values", _int, "cm/s");
    CGFEditorUtilities.BuildFloat("Float", "All float values", _float, _enum, "cm/s");
    CGFEditorUtilities.BuildFloatPositive("Positive Float", "Only positive float values", _float, _enum, "cm/s");
}
```

Boolean

Each property construction method has an overload with an additional parameter to add a boolean that will lock/unlock the property in the inspector.

In this example the property “_bool” will be the property locker, as we see in the code, every method adds as a parameter the the property “bool”.

```
public override void OnInspectorGUI()
{
    CGFEditorUtilities.BuildBool("Boolean", "Locker Boolean", _bool);
    CGFEditorUtilities.BuildInt("Integer", "All integer values", _int, "cm/s");
    CGFEditorUtilities.BuildIntPositive("Positive Int", "Only positive integer values", _int, "cm/s");
    CGFEditorUtilities.BuildFloat("Float", "All float values", _float, _bool, "cm/s");
    CGFEditorUtilities.BuildFloatPositive("Positive Float", "Only positive float values", _float, _bool, "cm/s");
}
```