Zones Download

Trigger zones and collider visualization for Unity projects.

Author: Igor-Valerii Chebotar **Email:** igor.valerii.chebotar@gmail.com

Requirements

- Sirenix - Odin Inspector
- Simple Man - Utilities

How to install plugin?

Open installer by the click on Tools -> Simple Man -> Master Installer -> [Plugins' name] -> Click 'Install' button. If you don't have one or more of the plugins this plugin depends on, you must install it first.

Quick start

1. Install
2. Add the standard zone to scene by right click inside the 'Hierachy' window -> Zones -> Event zone.
3. This zone will react on any object got inside. Use events 'On object entered' and 'On object came out'.
4. Done!

Custom zones

Here's how to create a new custom zone that will only respond to the "*CharacterLocomotion*" component and disable the ability to jump. Example and first person character asset you can found in 'Demo' package.

1. Create new C# class and inherit it from the 'Zone' class. Don't forget about 'using'. Add '*SimpleMan.Zones>*' and '*SimpleMan.FirstPersonCharacter*' namespaces.
2. Put your target component class in triangle brackets

```
using UnityEngine;
using SimpleMan.Zones;
using SimpleMan.FirstPersonCharacter;
```

```
public class NoJumpZone : Zone<CharacterLocomotion>
{

}
```

3. Override '*ValidObjectEnteredHandler*' and '*ValidObjectCameOutHandler*' methods.
4. Those methods will be called after object with component '*CharacterLocomotion*' entered / came out from zone. *You can also override '*CanBeRegistered*' method for creating custom registration condition: for example if your zone should react only on character with '*Health*' parameter value less then 50. All registered objects you can get by '*RegisteredObjects*' property, if you need to make operations with multiple objects inside the zone.
5. Argument '*collider*' is collider that was detected by zone. Argument '*component*' is target component on the game object. In our case component is '*CharacterLocomotion*'.
6. Make operations with '*component*' argument inside this methods.
7. Done!

```
using UnityEngine;
using SimpleMan.Zones;
using SimpleMan.FirstPersonCharacter;

public class NoJumpZone : Zone<CharacterLocomotion>
{
    //Object with 'CharacterLocomotion' component entered in zone
    protected override void ValidObjectEnteredHandler(Collider collider, CharacterLocomotion comp
    {
        //Forbid jump ability
        component.JumpForceMultiplier = 0;
    }

    //Object with 'CharacterLocomotion' came out from zone
    protected override void ValidObjectCameOutHandler(Collider collider, CharacterLocomotion comp
    {
        //Allow jump ability again
        component.JumpForceMultiplier = 1;
    }
}
```

How to add custom zone to the hierarchy create menu?

1. Save your custom zone as prefab in any folder.
2. Go to Edit -> Project settings -> Zones

3. Add your prefab to the list
4. **ALERT! Save all your unsaved scripts. Make sure, that you have't compilation errors before click on the 'Apply' button.**
5. Click 'Apply' button.
6. Wait for end of compilation process.
7. Done!

Zone component

Base component of all zones. Inherit your custom zone classes from it.

Properties

Property name	Description
UseTags	Enable tag checking toggle
AllowedTags	Objects with this tags only will be registered
RegisteredObjects	Readonly list of objects inside this zone

Events

Event name	Description
OnObjectEntered	Called on object was entered in zone
OnObjectCameOut	Called on object came out from zone

Methods

Function name	Description
ValidObjectEnteredHandler	Called on object was entered in zone
ValidObjectCameOutHandler	Called on object came out from zone
CanBeRegistered	Override it if you need to make a custom registration condition

Visualizer

Draws collider borders and labels. Can be used without zone component.

Properties

Property name	Description
DrawShape	Visualize collider borders toggle
WireShape	Draw collider borders by lines
DrawLabel	Draw label toggle
LabelSize	Size of the label text
LabelOffset	Vertical label text position offset
LabelSizeDependsOnScale	Should label change size with game object scale toggle
ShapeColor	Color of the collider borders
LabelColor	Label text color
ColliderTarget	Cached target collider

Collider switcher

Gives ability to change collider on the game object by one click in the inspector.

Properties

Property name	Description
CurrentCollider	Collider type