

# ISOMETRIUM PRO

# DOCUMENTATION

## v0.3beta

**Note:** This guide was written by an Italian.  
Sorry for the bad English, I hope that the concepts are clear.

**Isometrium** is an essential kit for those who want to create games with isometric style. Hack & Slash like Diablo / Torchlight / Sacred, isometric RPGs style, like Dragon Age or RTS games like Starcraft / RedAlert / AgeOfEmpire. But also action games like Borderlands Legends and more generally all top-down games and all games with a point&click system.

There are dozens of configuration combinations for the point image and mouse cursor, by the tag, by if an object/area is walkable, by additional scripts on objects...

Isometrium is very flexible and the sheer amount of options and configurations makes learning Isometrium quite challenging.

But do not worry, this document will clarify everything. If you have other questions please ask the author (preferably by email).

We are still in a beta version of the asset, and many features have yet to be implemented.  
Let's see what is already present in this beta.

### Scripts and prefabs



**Isometrium (Main)** (the Main prefab to add into your scene)



**Isometrium(MousePoint)** (a children prefab under Isometrium(Main))



**Isometrium (MouseCursor)** (a children prefab under Isometrium(Main))



**Isometrium(Selections)** (a children prefab under Isometrium(Main))



**IsometriumCamera** (a script to add to your main Camera)



**IsometriumCursorOnObject** (a script to add to your gameobjects)



**IsometriumPointOnObject** (a script to add to your gameobjects)



**IsometriumSelectableObject** (a script to add to your gameobjects)



**IsometriumClickableObject** (a script to add to your gameobjects)



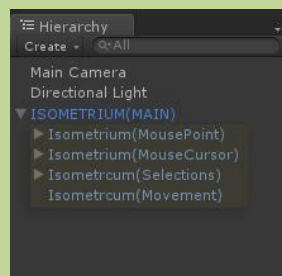
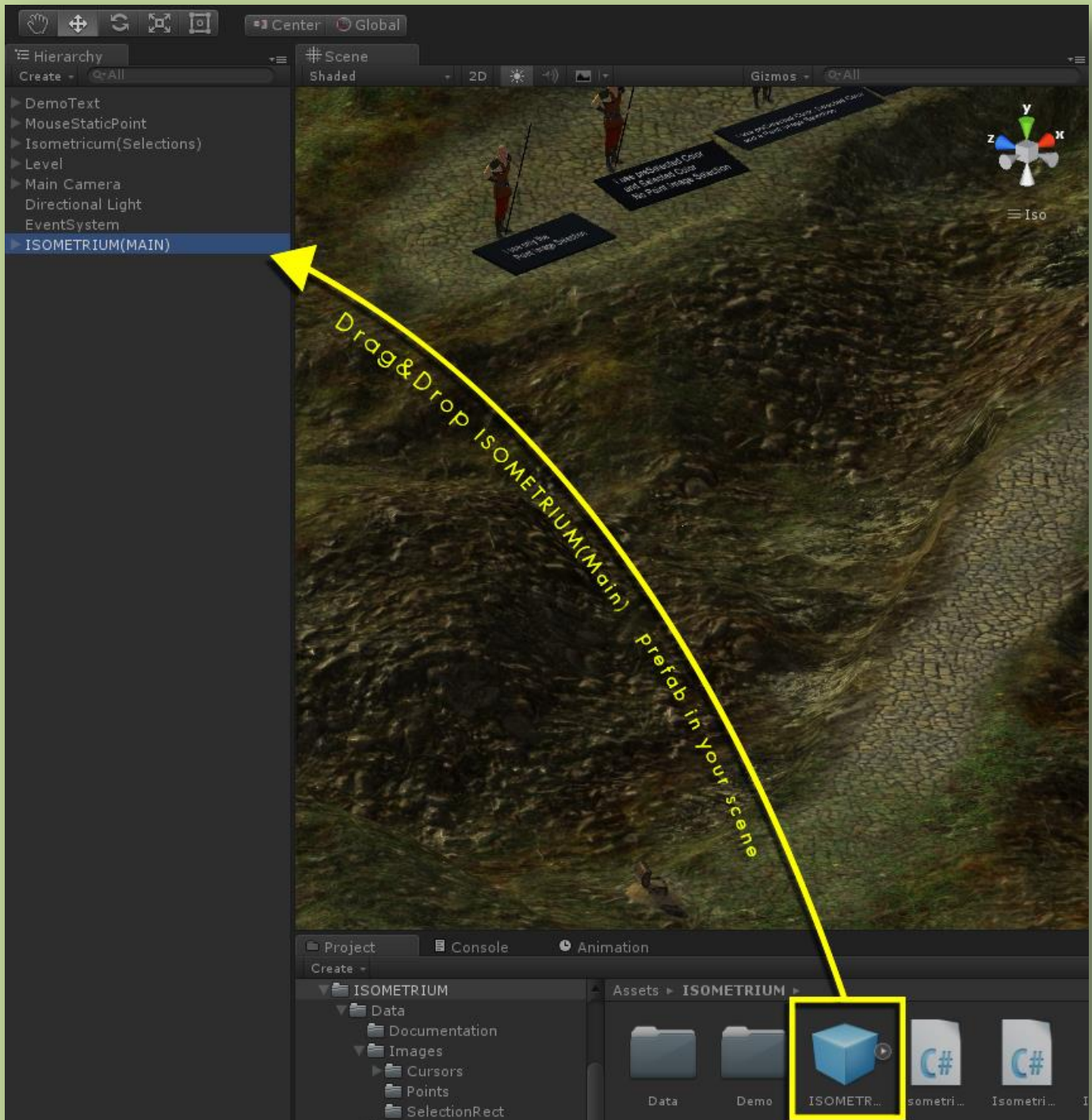
**ISO\_TagDependence** (a script to add to your gameobjects)

Small legend and abbreviations of that can be found in some ISOMETRICUM components

ABBREVIATION	MEANING	SUMMARY DESCRIPTION
ISO	IsometriumSelectableObject	Script to be added to RTS units, members of a RPG party or to RPG player...
ICO	IsometriumClickableObject	Script to be added to RTS enemy units, RPG enemy, collectible items, RTS resources etc ...
	MousePoint	This term shall understand the 2D image/icon projected over the terrain and over objects.
	No Target Point Over This	This bool can refer to a tag or to an object. Basically, the target point is no placed on an object with this bool checked. It is useful if an object is not walkable.

To use Isometrium, simply drag&drop  
ISOMETRICUM(Main) Prefab in your scene

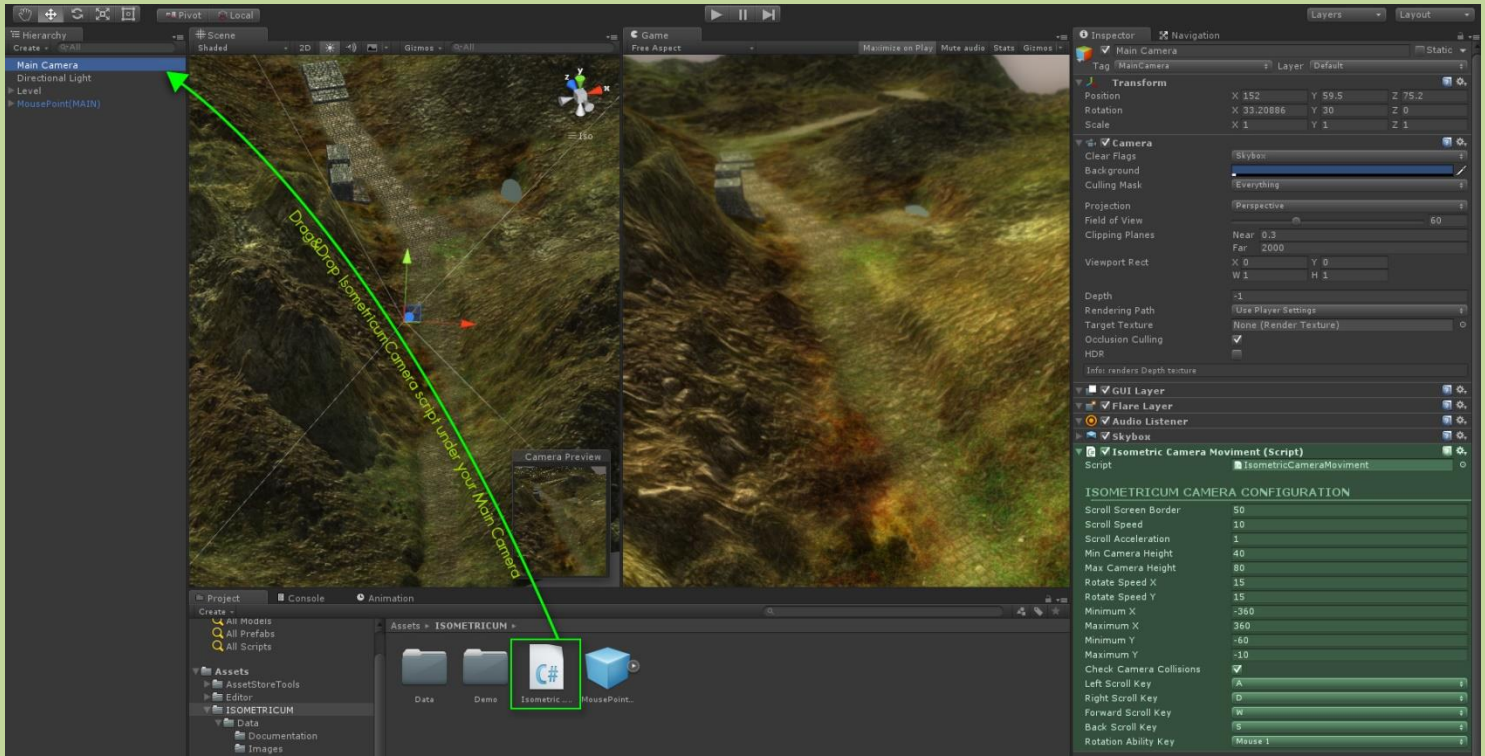
Only this prefab is strictly necessary for the operation of ISOMETRICUM.



Now you can configure 4 scripts in childrens,  
**MousePoint, MouseCursor, Selections and Movement.**  
(Please, do not change other settings in the other childrens  
scripts)

But before of that, **add Isometrium camera** to you scene.

# IsometriumCamera



IsometriumCamera is a script (component) to add to you Main Camera.

With this component you can configure the camera and his way of moving.

The variables to be modified are all very intuitive and do not require detailed descriptions.

**Screen Scroll Border** sets the width/height edges near the screen borders where mouse position shall start to slide the camera.

You can visualize this areas by **Visualize Border At Runtime** var.

If you do not want to implement this feature (you can still use the keyboard keys to move the camera) set this value to 0.

If you want a Diablo III style camera (without camera rotations), set to "None" the variable "Rotation Ability Key".

If you do not want to scroll the camera with keyboard keys, put "none" the four parameters that move the camera.



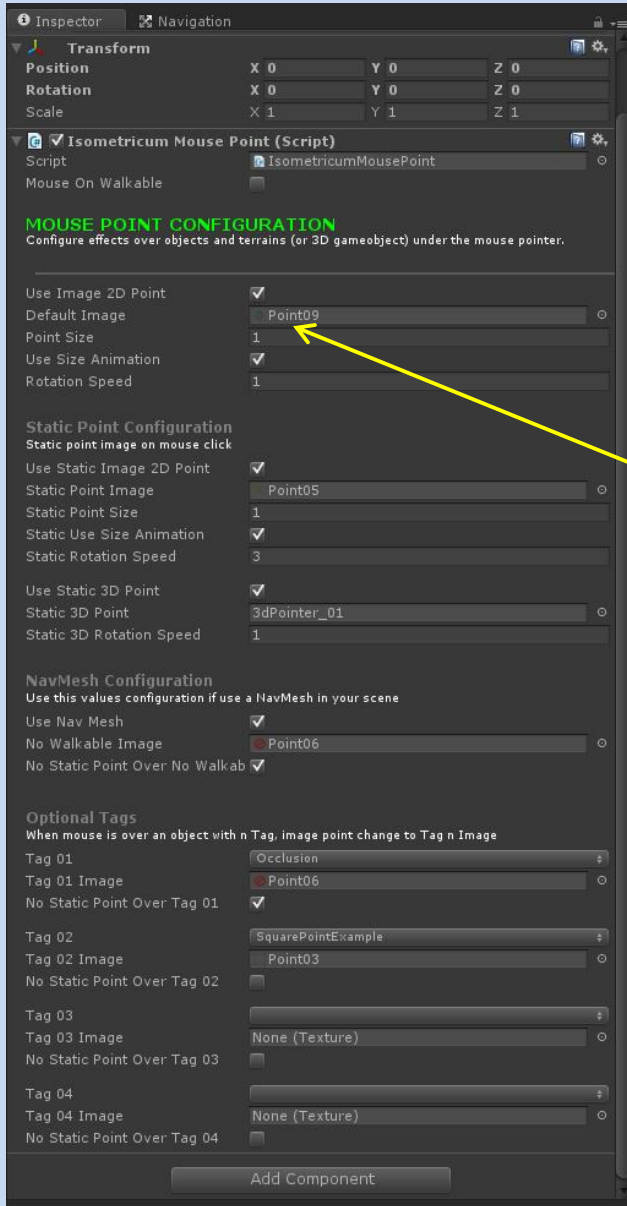
Hold Down the Double Scroll Speed Key to double the Scroll Speed.



# Isometrium(MousePoint)

**Isometrium (MousePoint)** is a component to configure the images point below the mouse that will appear over the objects and over the ground.

This image changes depending on the tag of the object under your mouse pointer or by NavMesh area (walkable or not walkable), if any.



It isn't the mouse pointer. The management of the mouse pointer is on another script (Isometrium(MouseCursor)).

**Isometrium (MousePoint)** is the projection of an image (a texture2D) over the geometry underlying the mouse pointer.

This is the an image in the demo.

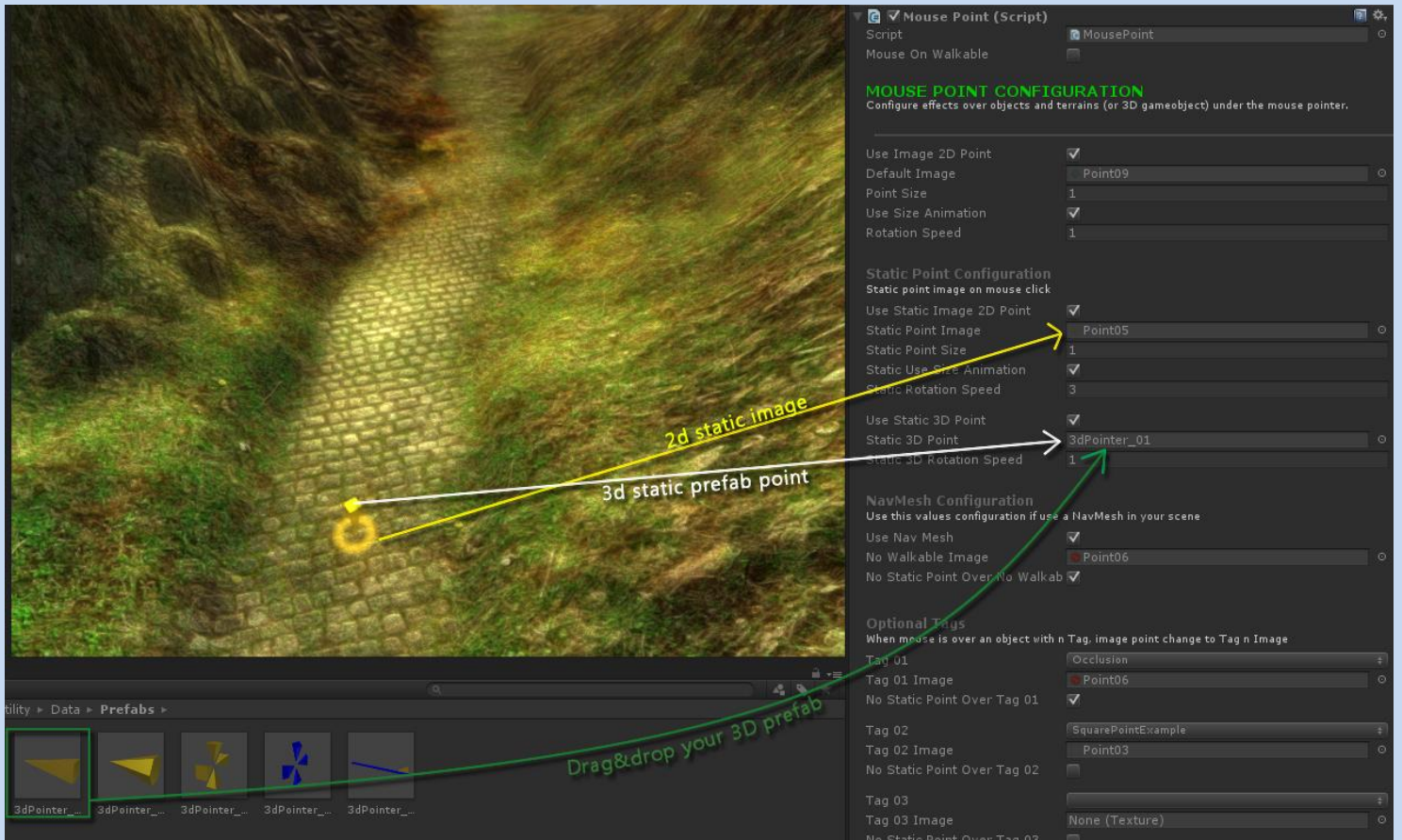


There are many images already in asset and many more will be included (also available on request).

You can set any image created by you.

You can set the size, a loop animation and a continuous rotation. Set to 0 "Rotation Speed" if you do not want to rotate.

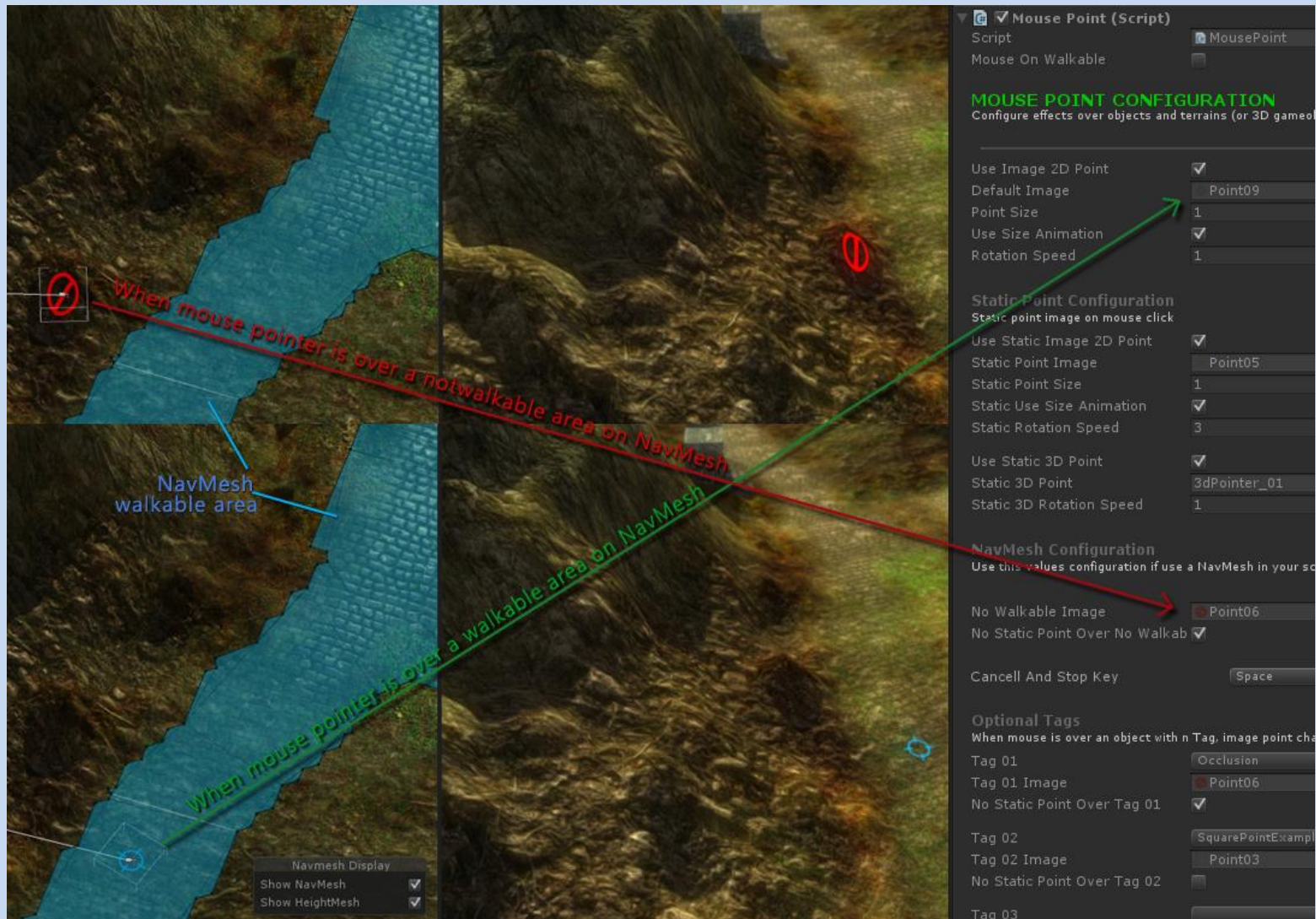
**Target Point** (named *StaticPoint* in some old shots) is the image/gameobject that appears at the mouse click on a walkable area. You can choose to use a 2d image only, a 3D point only or both simultaneously.



Again, you can choose the size, whether to rotate the point and also scale animating. There are several 3d prefabs in asset and many more will be added (also available on request).



If in your scene uses a **navMesh** can use the next feature: **NavMesh Configuration**



It's a simple but very powerful feature!

You can choose which point image appears when the cursor is on a NavMesh not walkable area.

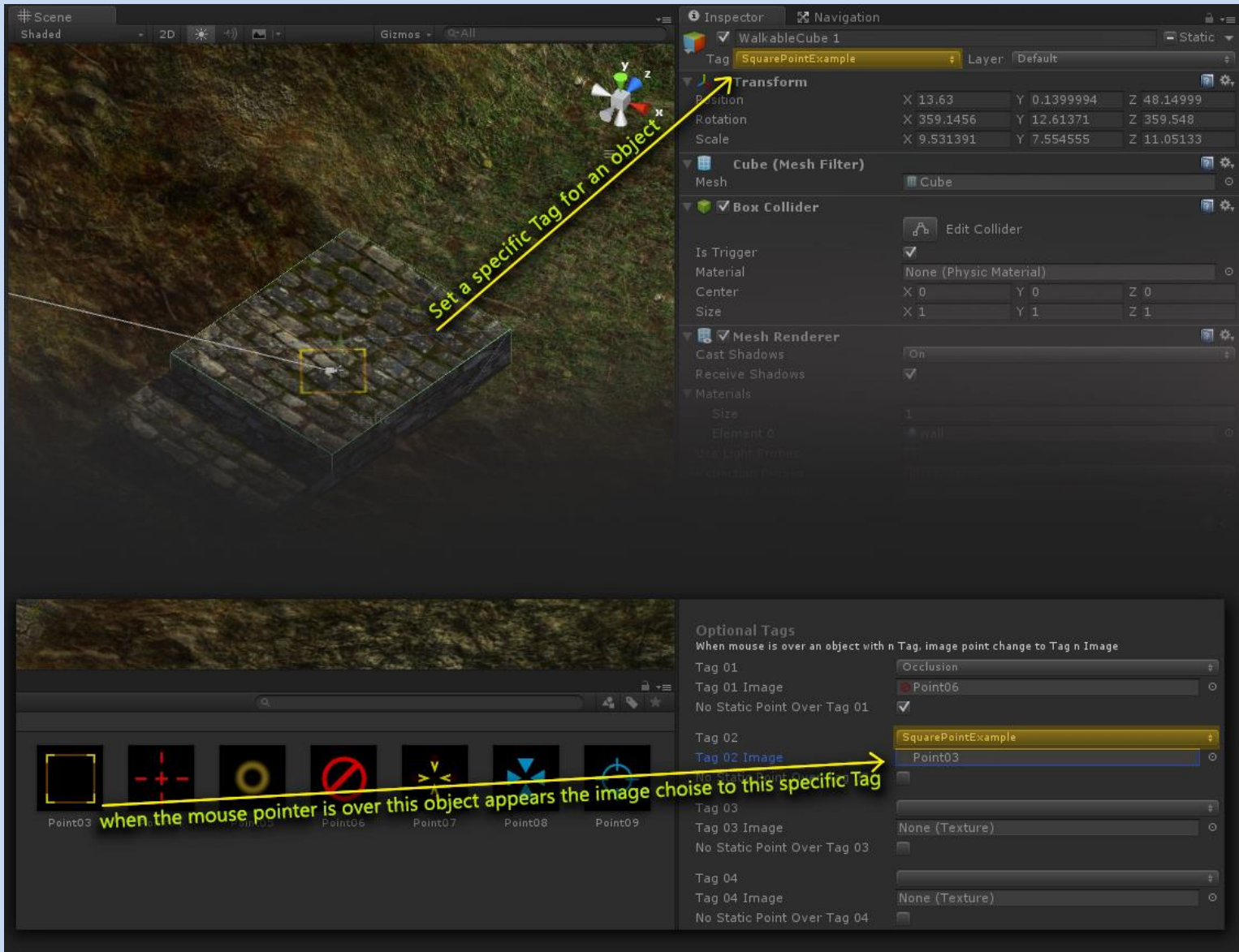
The "No Target Point Over No Walkable" option disables the target point over the "Not Walkable" NavMesh layer.

In practicality this means that clicking on a point in "Not Walkable" area will *not* perform an action.

With **Cancel ad Stop Key** you can choose which button will call `StopSelectedUnits ()` function.

This function stop all selected units and remove its target points.

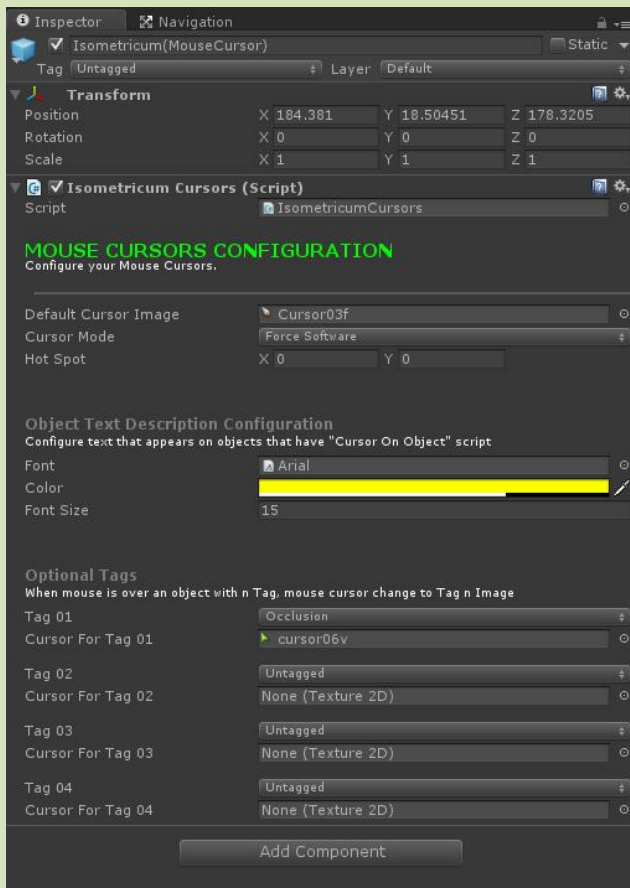
**Isometrium (MousePoint) Optional Tags** is a section that allows to put 4 specific Tags to be able to use on objects to diversify the points images that appear on them.



For Tag 01 corresponds the Tag 01 Image, for Tag 02 corresponds the Tag 02 Image and so...  
For each of them you can choose the "No Target Point Over Tag n" option which functions as explained above, clicking on a gameobject with that Tag will not perform an action.



# Isometrium(MouseCursor)



With this **prefab** you can choose the default mouse cursor and 4 other cursor that changes depending on the object tag on mouse pointer is over.

You can also configure the font, text color and font size of the text, that appear under the pointer. We will see this later.



There are two way to change cursor depending on the object on mouse pointer is over:

- 1) Use this 4 **Optional tags**.(work just like for the **Isometrium (MousePoint)** optional tags).
- 2) Use the script **CursorOnObject** on every single object you want together to **ISO\_TagDependence** script.

We will see them in more detail below.



**The Isometrium (MouseCursor)** optional tags are useful if you have many objects of the same type to which you want to apply the same cursor.

For example, if you have collectible objects in your game, you can apply a tag like "collectible" and a like 🖱 mouse cursor associated with this tag.

So for all items tagged "collectible", the 🖱 mouse cursor appears when you pass the mouse is over them.

A easy and very fast solution, but limited, however, are only 4 and may not be enough. For this there are **CursorOnObject** and "" scripts described a little below.

# IsometriumSelectableObject (ISO)

When an object has this script can be selected. Abbreviated to "ISO". **IsometriumSelectableObject** is the script to add to the NPC units as the units of an RTS or to Hack & Slash player.

**Object Name or Type** is an optional string for an additional unique name to identify this object or a generic string to identify a type of unit (Like soldier, civilian, workers etc...). If not inserted will be used the name of this gameobject.

This string will be useful for automatic selection of all units with the same parameter by double-clicking on one of them.

For example if you have many units with this variable "riflemen", double click on one of them will be selected all objects with the same string.

**Mintain Its Target Point** is a useful bool var.

If cheked, when you mouse over on this IsometriumSelectableObject (or this is selected) you will display its current TargetPoint (if any).

For example, if you sent a unit on a point of the map or to attack an enemy or to collect a resource... and after you've done something else... at any time you can pass your mouse over this unit you will

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

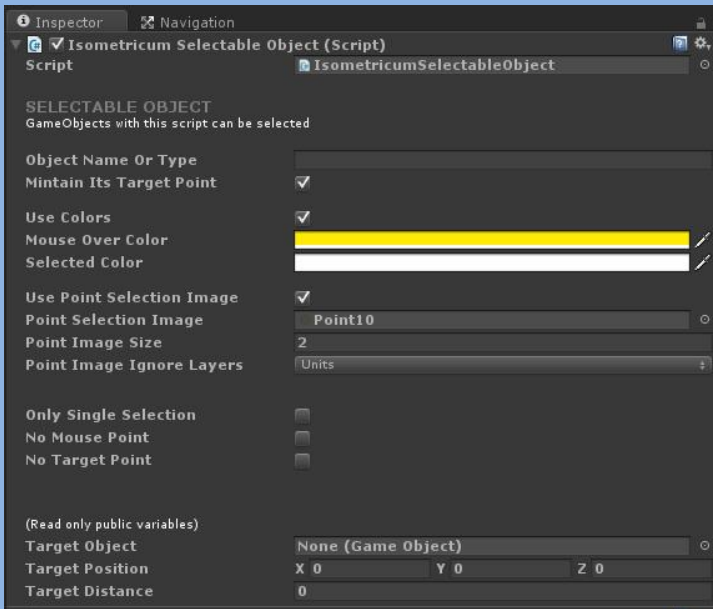
see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

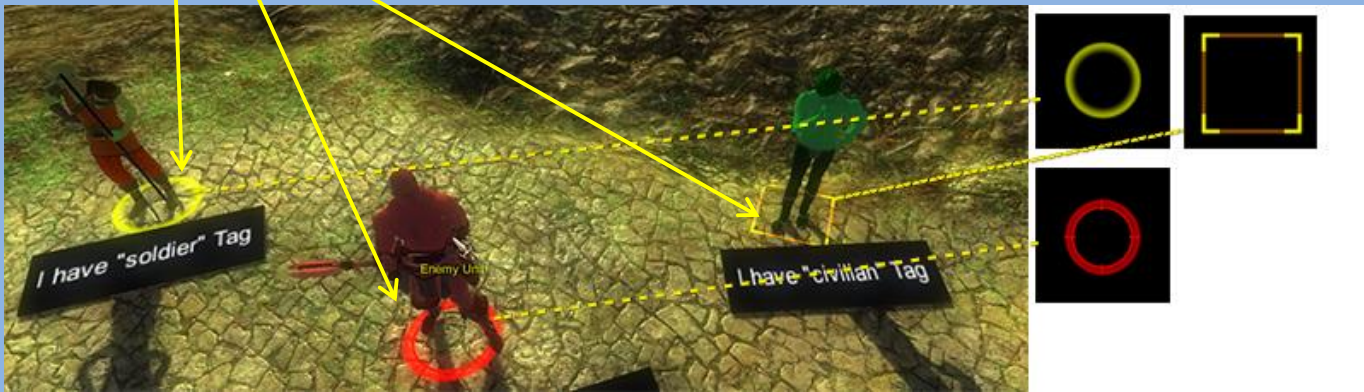
see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.



see the point where it's going.. In practice, this bool remember you what it is doing and on which object or map point is directed at that time.

**Use Colors** determines the use of colors for the mouse-over and object selection.

**Point Selection Image** is the image under the IsometriumSelectableObject when it is selected (on mouse-over for the IsometriumClickableObject like enemies that we will see a little later).



You can set the image (a texture with clamp warp mode), the scale and Ignore Layers.

**On Point Image Ignore Layers**, this Point Image image will not be projected. If you do not want be projected the image onto the object itself, enter also the layer of belonging the gameobject.

The following three variables are closely linked to each other.

**Only Single Selection** is a bool to ensure that this object is selectable only alone.

If checked, this SelectableObject ignore the Rectangle Box Selection and can only be selected with a mouse click.

Multiple selection will not work with a SelectableObject with this bool checked, and it can only be selected alone and not together with other SelectableObject. From it depend the following two bool.

**-No Mouse Point**: if checked, also when this SelectableObject is selected do not show any projected image under mouse pointer. This option works only if OnlySingleSelection bool is cheked!

**-No Target Point**: if checked, this SelectableObject do not show target point on mouse click. Also this option works only if OnlySingleSelection bool is cheked!

These options are useful for some particular objects or human units. They can also be used for objects such as simple buildings.

**Read only public variables** are useful to know the information on this SelectableObject. Should not be inserted from you.

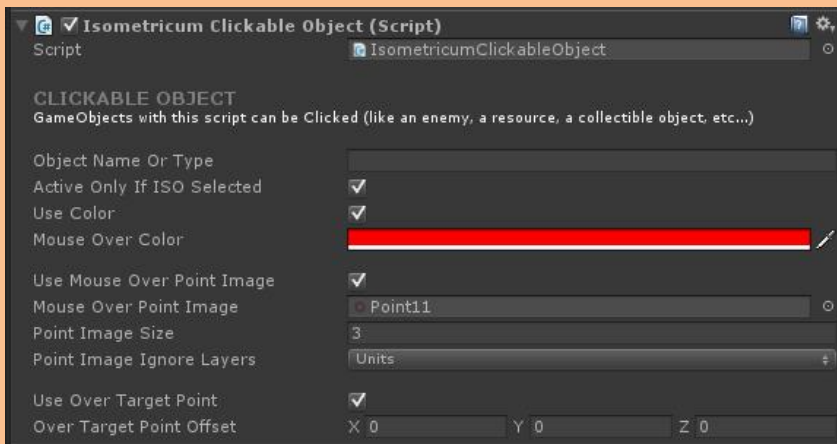
**Target Object**: this is the IsometriumClickableObject on which you click when this IsometriumSelectableObject is selected (like an enemy to attack, a resource or object to be collected...)

**Target Position**: is the Vector3 point on which you click when this IsometriumSelectableObject is selected (like a walkable area or IsometriumSelectableObject position)

**Target Distance**: is the tistance between this IsometriumSelectableObject and the TargetPosition (Updated in real-time).

# IsometricumClickableObject (ICO)

GameObjects with this script can be clicked and not selectable, like enemies units to attack, collectible objects, resources etc... . Abbreviated to "ICO".

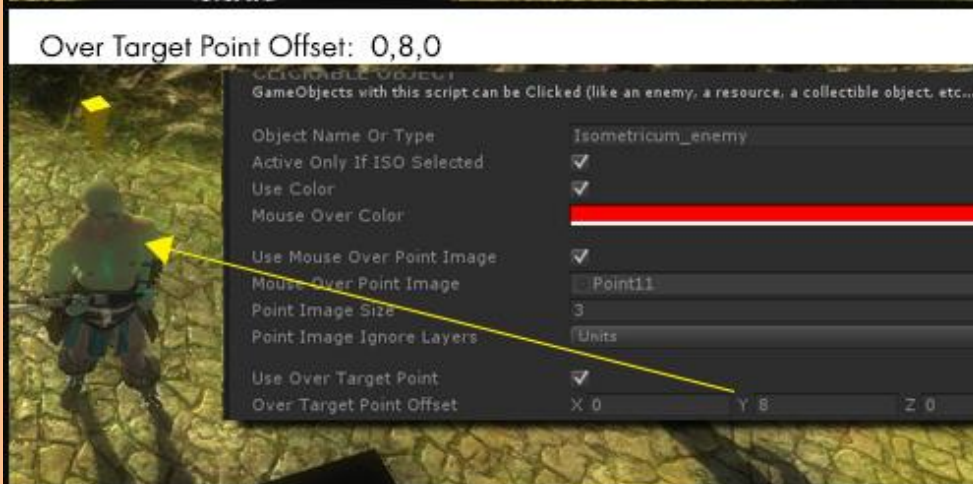
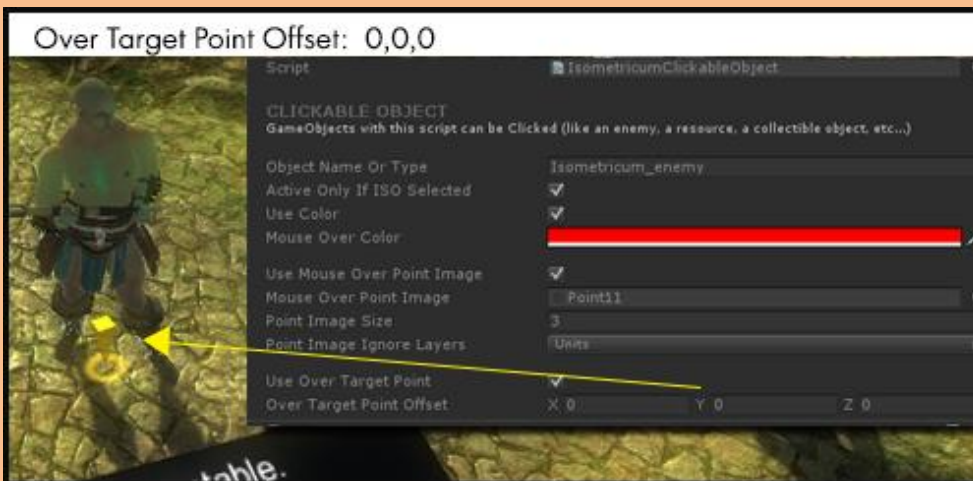


Most of the editable options are the same as ISO script with the difference that colors and the Point Image applies only to the mouse-over.

**Active Only If ISO Selected:** If checked, the mouse over will be active only if at least one ISO it is currently selected.

**Use Over Target Point:** If checked, when a ISO is selected and you click on this object, you will see the ISO target point over it.

**Over Target Point Offset:** Is a Vector3 variable to set and adjust the **Over Target Point** position. If set to 0,0,0, the **Over Target Point** will be placed on the origin of gameobject.





## A bit of code

**IsotriumManager** is the Main Class where you can access to most of functions and variables. It is a class that will be automatically added to the scene.

```
//Return a String. The tag of object under the mouse pointer  
IsotriumManager.MouseOverTag  
  
//Bool var, true if mouse pointer is over a NotWalkable NavMesh Area  
IsotriumManager.OverNoWalkableNavPoint  
  
//Bool var, true if mouse pointer is over a Walkable NavMesh Area or Walkable object  
IsotriumManager.MouseIsOnWalkable  
  
//Metod to show the Target Point  
IsotriumManager.ShowTargetPoint ()  
  
//Metod to hide the Target Point  
//Use this function to hide the Target Point when you want  
IsotriumManager.HideTargetPoint ()  
  
//Call this function to Stop all selected units and remove its target points  
IsotriumManager.StopSelectedUnits ();
```

Documentation under construction....

For info or request: [wilez@wilez.it](mailto:wilez@wilez.it)