Hierarchy Explorer Documentation

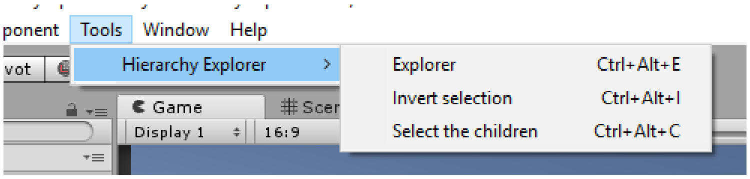
Wawrzyn321

1 About the project

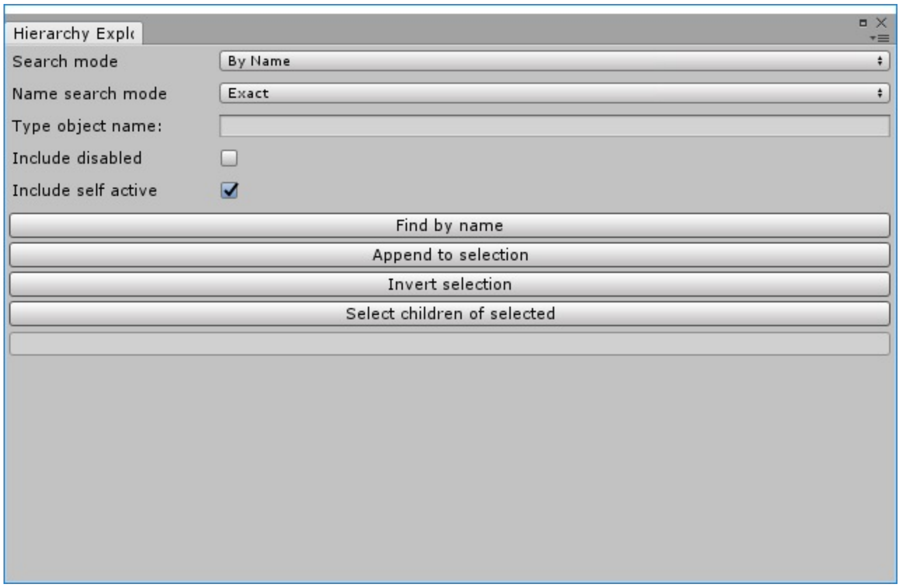
This little Unity Editor extension is made to facilitate the Hierarchy searching. It provides the user extended ability of searching in project's hierarchy, like appending found enitites to already selected objects, ignoring the case of search and differentiation of active and inactive objects. It makes it possible to find objects by tags or types of their components.
Note that actions performed within the Hierarchy Explorer's window may deselect objects in Project tab.
Note 2: Remember that some tags and layers are imported to project for testing usage. It's safe to remove them.

2 Window overview

To begin, select the "Hierarchy Explorer" from "Tools" bar.



The following window will appear:



Search mode: By Name

Name search mode: Exact

Type object name:

Include disabled □

Include self active ☑

Let's go through all the options:

2.1 Search mode

This dropdown gives us option to choose the way of finding the object: by name, tag or layer. The options below will change accordingly.

2.2 Name search mode

2.2.1 Exact

Return object only and only if it's name is equal to typed name.

2.2.2 Case insensitive

Like exact, but this time ignore the case.

2.2.3 Simple Regex

Use '?' character as wildcard for a single character in looked name. End-trailing '?' are ignored, so 'S???' will return 'S1', 'S11', 'S111', and not 'S1111'.

2.2.4 Regex

Use C# System.Text.RegularExpressions.Regex.

2.3 Text field

Here you can type the property of group of objects you want to find.

2.4 Enability options

2.4.1 Include disabled

If off, objects that are inactive in hierarchy are excluded from search.

2.4.2 Include self active

If on, object that is *activeSelf* will be included in search results, even though it can be child of inactive parent.

2.5 Find & Append

Find option replaces already selected entities with objects found. If none was found, the selection is left intact. In turn, the *Append* button only adds found object to current selection content.

2.6 Invert selection

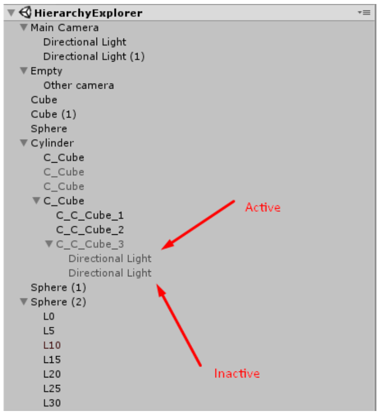
Replaces the selection with collection of objects that weren't selected at the time. Of course, the enability options apply here.
As you can see, this action can be accessed by menu or by shortcut. Note that when performed this way, it will affect only the active objects.

2.7 Select children of selected

Iterates through already selected objects and adds their children to selection. As while inverting the selection, the enability rules are also applied here and it also can be invoked by shortcut.

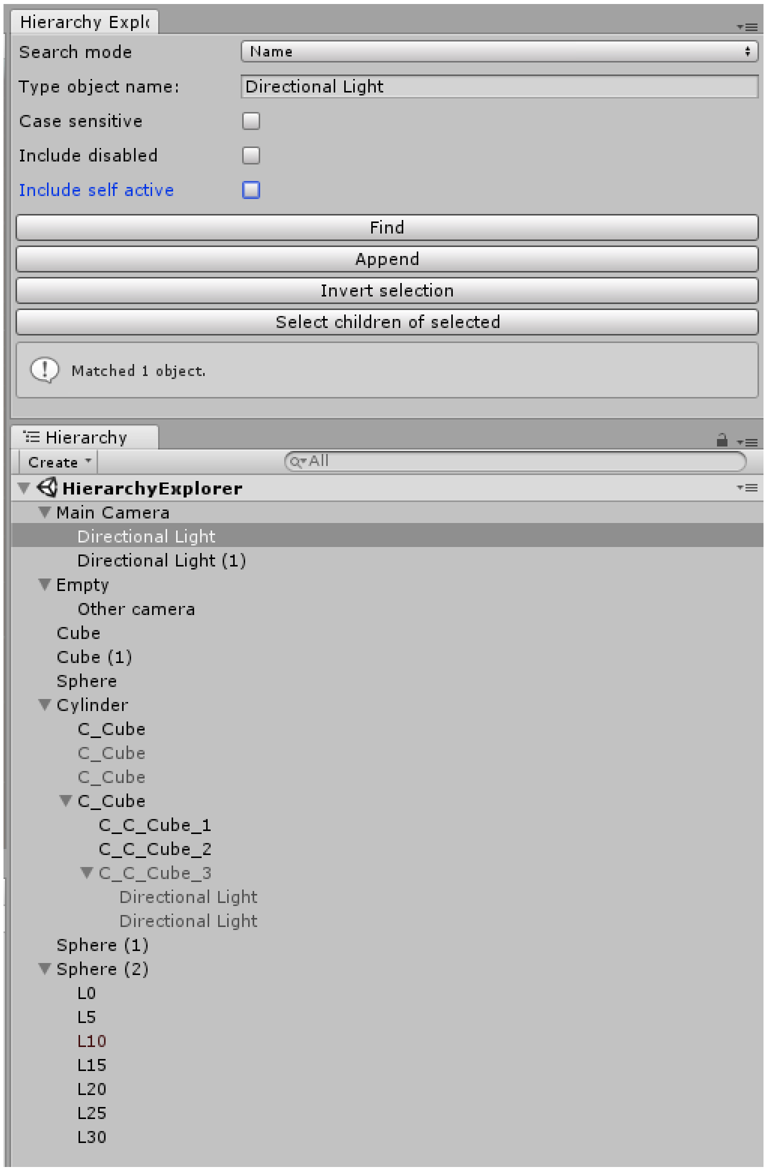
3 Detailed differences between enability modes

Let's elaborate about enability options. In asset's unitypackage, there is a HierarchyExplorer scene with exemplary hierarchy.



As you can see above, both pointed Lights have different options - first is enabled, the second has been disabled in Inspector - yet, they are both inactive because their parent (C_C_Cube_3) is inactive itself.
Let's run Hierarchy Explorer with "Directional Light" type and mode set to *Name*.

3.1 Don’t include disabled nor *activeSelf*



Search mode: Name

Type object name: Directional Light

Case sensitive □

Include disabled □

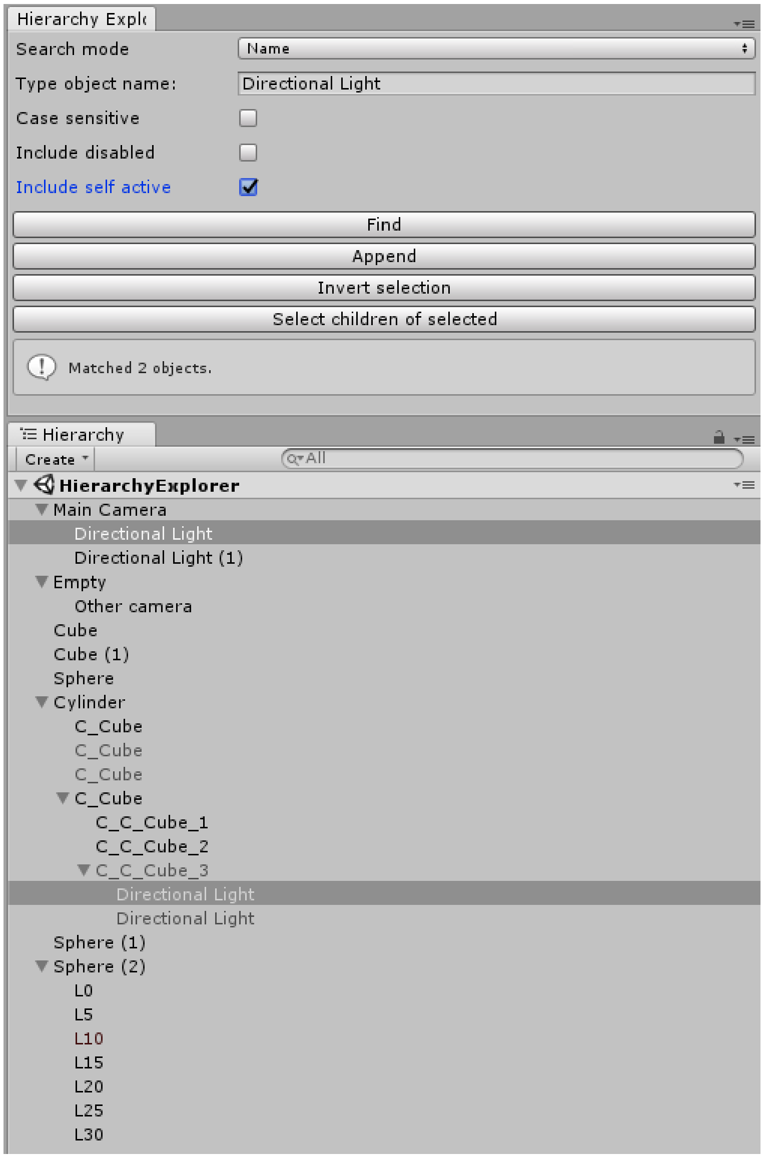
Include self active □

Matched 1 object.

HierarchyExplorer
- Main Camera
  - Directional Light
  - Directional Light (1)
- Empty
  - Other camera
- Cube
- Cube (1)
- Sphere
- Cylinder
  - C_Cube
  - C_Cube
  - C_Cube
  - C_Cube
    - C_C_Cube_1
    - C_C_Cube_2
    - C_C_Cube_3
      - Directional Light
      - Directional Light
- Sphere (1)
- Sphere (2)
  - L0
  - L5
  - L10
  - L15
  - L20
  - L25
  - L30

Only Directional Light on top has been selected, because the other two are simply inactive.

3.2 Don't include disabled, but let *activeSelf* pass



Hierarchy Explo

Search mode: Name

Type object name: Directional Light

Case sensitive □

Include disabled □

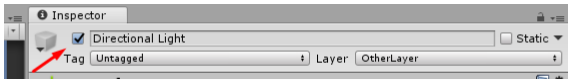
Include self active ☑

Matched 2 objects.

Hierarchy

```
HierarchyExplorer
  Main Camera
    Directional Light
    Directional Light (1)
  Empty
    Other camera
  Cube
  Cube (1)
  Sphere
  Cylinder
    C_Cube
    C_Cube
    C_Cube
    C_Cube
      C_C_Cube_1
      C_C_Cube_2
      C_C_Cube_3
        Directional Light
        Directional Light
  Sphere (1)
  Sphere (2)
    L0
    L5
    L10
    L15
    L20
    L25
    L30
```

Now first Directional Light is selected, because even if it isn't marked in black (so that it isn't technically active), the *activeSelf* checkbox is marked.

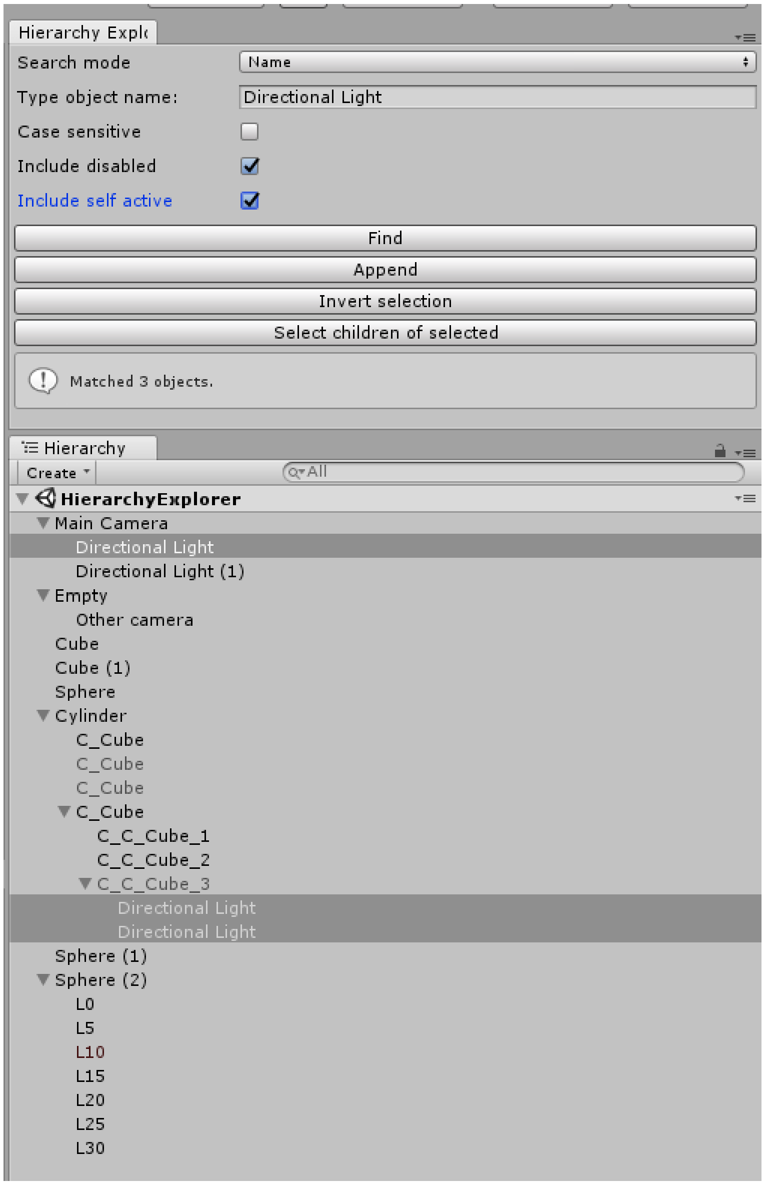


Inspector

☑ Directional Light □ Static

Tag: Untagged Layer: OtherLayer

3.3 Include disabled



Search mode: Name
Type object name: Directional Light
Case sensitive □
Include disabled ☑
Include self active ☑

Matched 3 objects.

```
▼ HierarchyExplorer
  ▼ Main Camera
      Directional Light
      Directional Light (1)
  ▼ Empty
      Other camera
    Cube
    Cube (1)
    Sphere
  ▼ Cylinder
      C_Cube
      C_Cube
      C_Cube
    ▼ C_Cube
        C_C_Cube_1
        C_C_Cube_2
      ▼ C_C_Cube_3
          Directional Light
          Directional Light
    Sphere (1)
  ▼ Sphere (2)
      L0
      L5
      L10
      L15
      L20
      L25
      L30
```

This one is simple - collect everything that meets the *Name* conditions, that is: both active and inactive objects. The *Include self active* option doesn't really matter now!

4 Contact

If you have any questions or issues, don't hesitate to contact me @ Wawrzyn321@gmail.com