

CSS-like Search (pro) manual

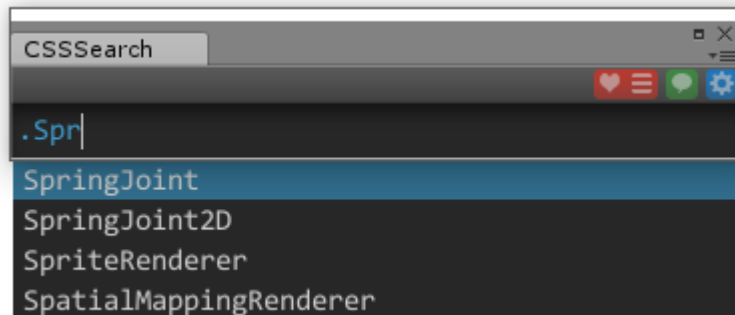
The advanced version of CSS-like Search includes all the features of free version, you can read about them in docs_free.pdf (read it first).

CSS-like Search (pro) additional features:

- Autocomplete (code completion while you type)
- Regex support for string attributes
- Favorite queries (save frequently used queries)
- Advanced attribute search (like `sprite.name!="tank"`)
- Search in assets

Autocomplete

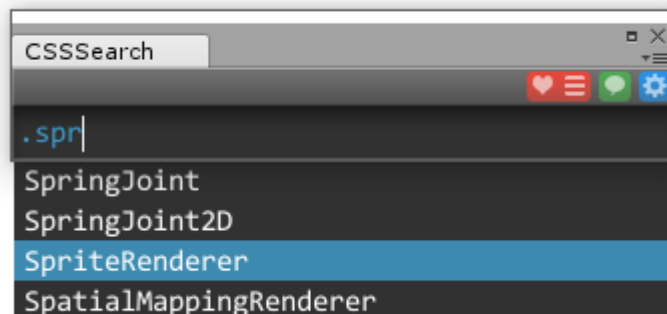
Helps you to build query you need faster and easier.



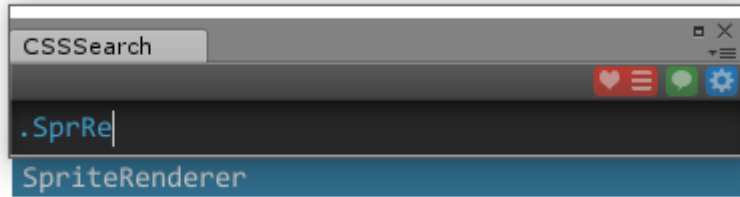
Suggestions are shown below the window. Class, name and attribute names are suggested, depending on selector. In given example matching components' names are shown for class selector.

Imagine, we want to enter SpriteRenderer.

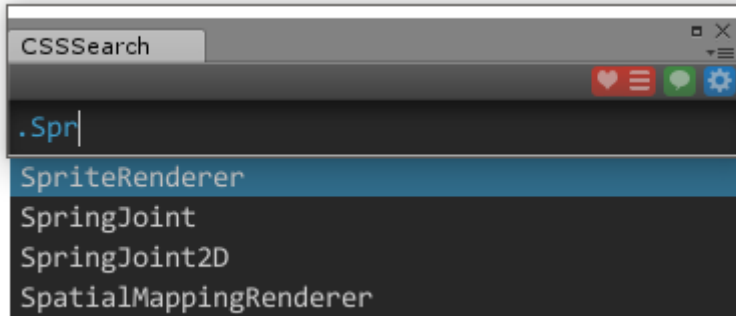
The first way, how to do this – is to move selection with arrow (up and down) keys and hit enter (or click by mouse).



The second way is to type part of the second word, so text will be "SprRe" and hit enter:

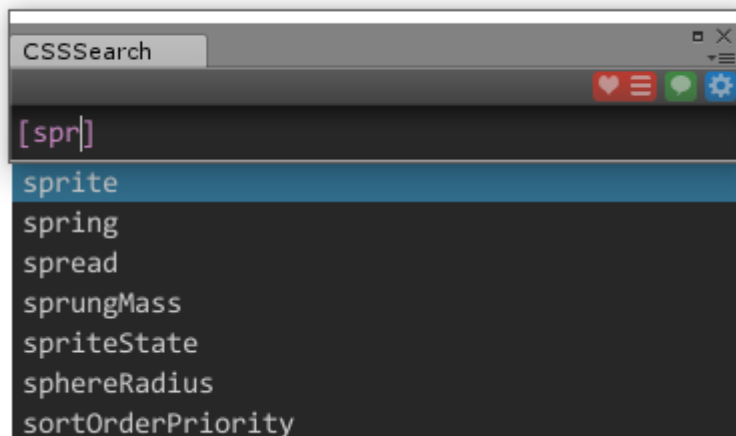
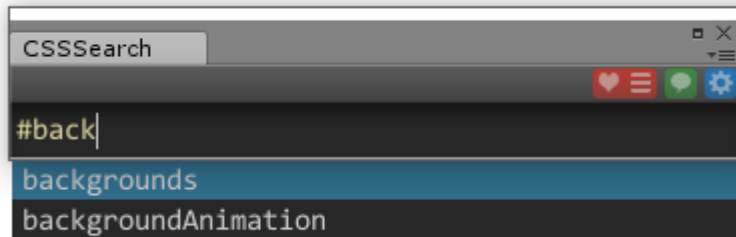


In case of frequent using SpriteRenderer, system will suggest it as a first item:



Autocomplete history is used for this. If you want to reset it, you can do it in preferences window.

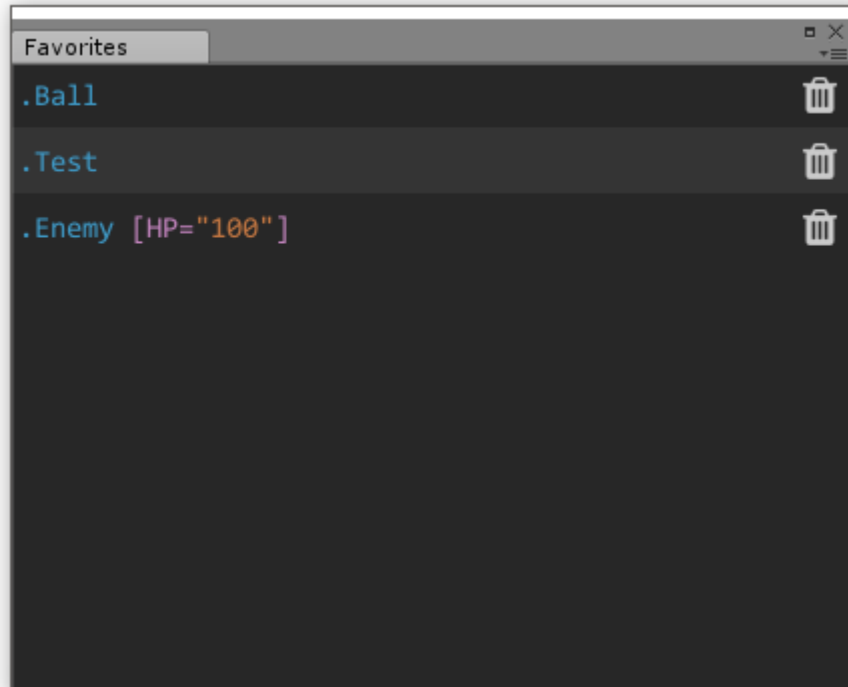
Name and attribute selectors are supported also:



Favorite queries

Some queries that are frequently used or complex to type may be saved to favorites. Click the like button and the query will be added to favorites.

To see the list of favorite queries, click the sandwich icon near the like button. Window will appear:

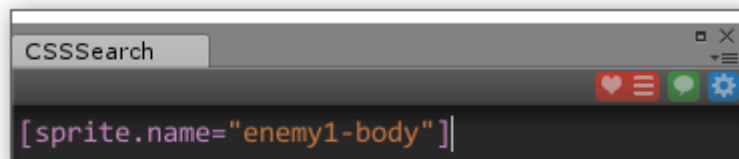


Here you can delete any of queries by clicking the trash icon.

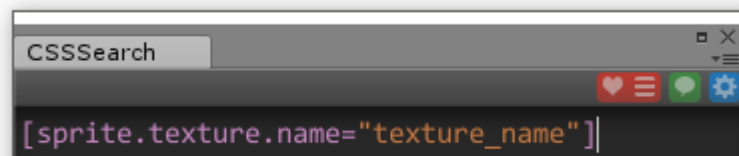
To use query, click on its item in the list. The window will close and query text will be inserted into search field.

Advanced attribute search

In attribute selector sequence of properties/field can be defined. Ex (all objects with given sprite name):



You may specify sequence of any number of elements:



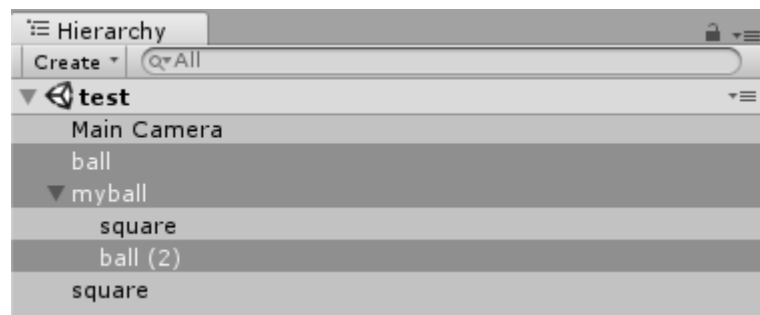
Additional operators are available:

Attribute type	Operators
Integral numbers	=, >, <, <=, >=, !=
Floating numbers	=, >, <, <=, >=, !=
Vector	=, !=
Enum	=, !=
Bool	=

Regex in attribute value

For string attributes, you can use regular expressions. Just add '\$' sign before string value and it will be parsed as regex. Full C# regex functionality is supported. Regex is automatically modified to have '^' sign (start of string) at start and '\$' sign (end of string) in the end, so attribute value is checked fully.

For example, you can select all objects that contain string:

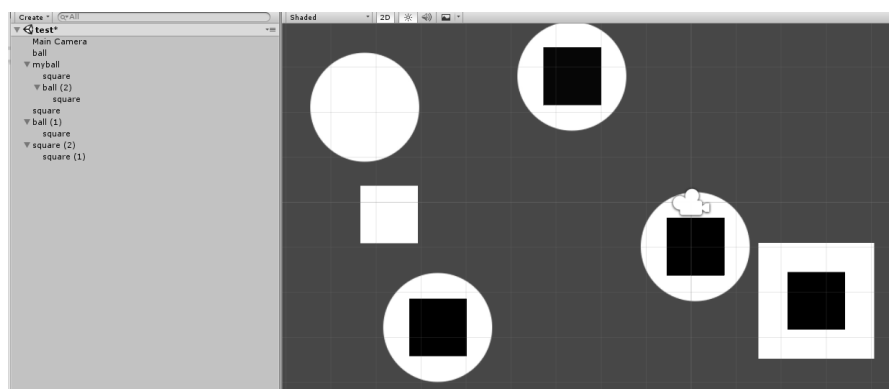


Parent selector

Pro version of CSS-like search includes additional selector – parent selector. There is no parent selector in CSS (maybe because it may make CSS inconsistent), but for our purposes it adds more abilities for search.

You can insert parent selector as "<" sign.

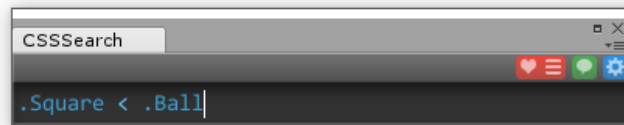
Let's consider given test scene:



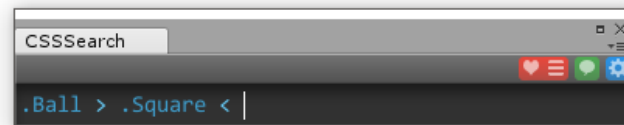
Each square and ball has “Square” and “Ball” scripts attached respectively.

The aim is to select balls that contains squares.

The first way to do this is to select all squares, then their parents and filter them by class (.Ball).

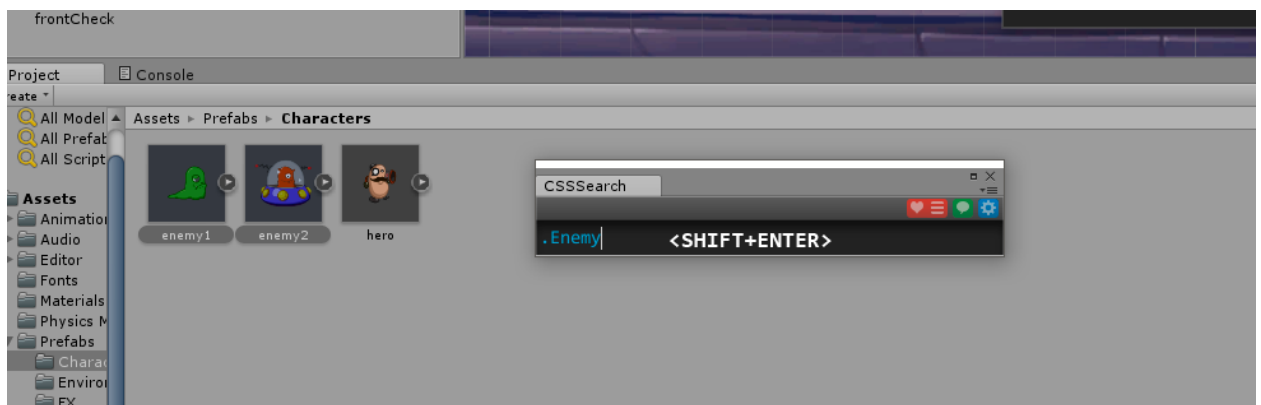


Another way to do this is to select all balls first. Then go to children, do all checks (in given example we just check there is a Square script), and then return back to balls with parent selector.

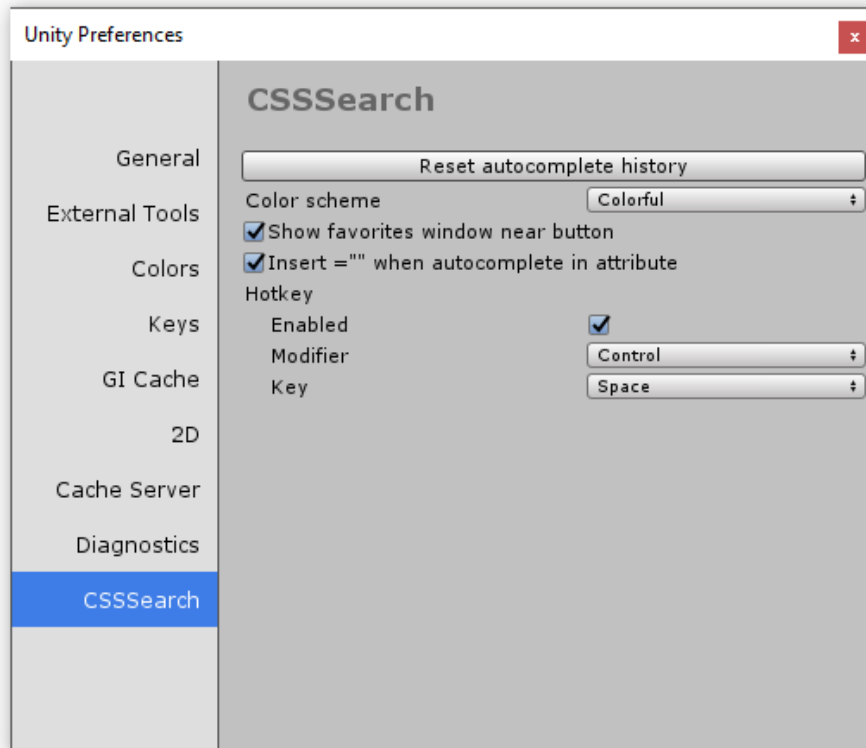


Asset search

You may execute search in assets, pressing shift+enter. All matching elements will be selected.



Preferences



- Reset autocomplete history – clears history, so predictions are not based on what you selected before
- Color scheme – if you don't like colored buttons – you can make them grey
- Show favorites window near button to allow you quickly select queries
- Insert ='' when you choose autocomplete item in attribute selector. Might be better to disable it if you frequently use sequence of fields/properties (like sprite.name)
- Hotkey – here you can enable/disable hotkey for popup. Also you can change modifier and key itself. Please choose hotkey that does not conflict with other Unity editor hotkeys