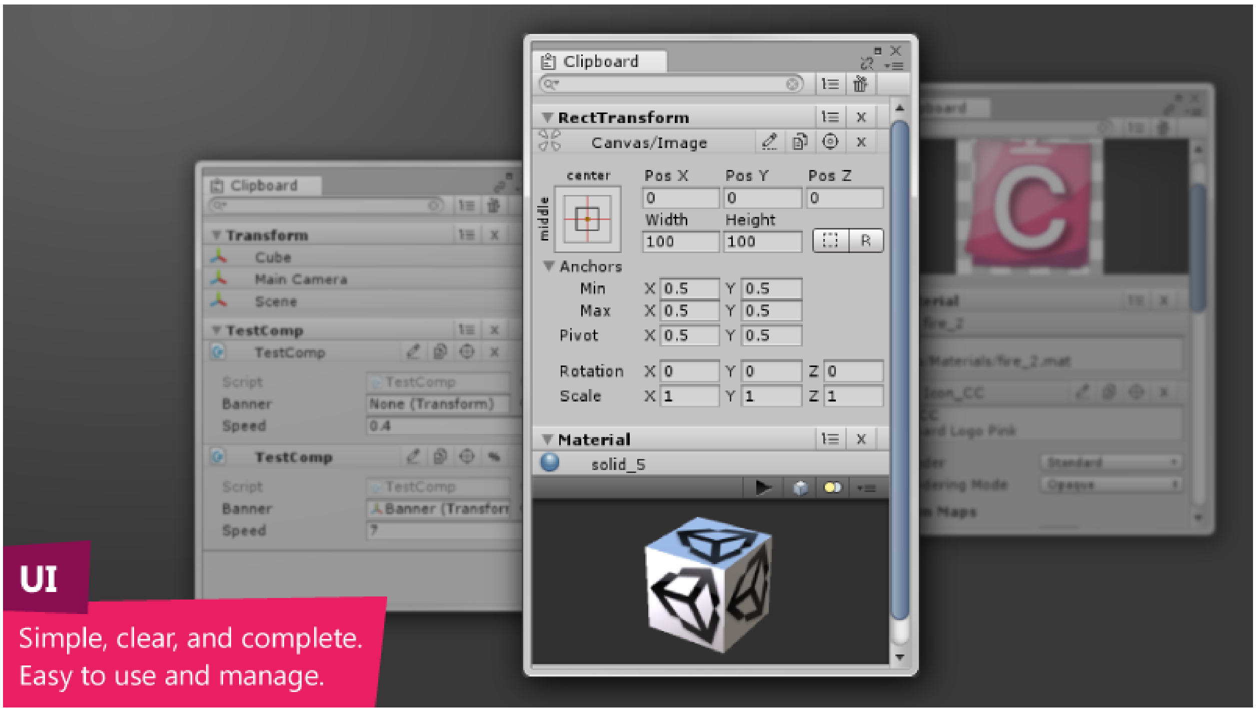
Component CLIPBOARD

Let's save your time from routine work.

- Have you ever tweaked lots of Components in play mode, and noted values for later?
- Do you need to try many settings or Material properties to pick which looks better?
- Do you even want to copy a lot, then paste after lunch or the weekend?
- Would you like a Component presets panel to apply values quickly?

That's why I made this!
Watch the video shows how it looks and works, and another show its smooth workflow.



Clipboard

RectTransform
Canvas/Image

center	Pos X	Pos Y	Pos Z
middle	0	0	0
	Width	Height	
	100	100	

Anchors
Min X 0.5 Y 0.5
Max X 0.5 Y 0.5
Pivot X 0.5 Y 0.5

Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Material
solid_5

Clipboard
Transform
Cube
Main Camera
Scene
TestComp
TestComp
Script TestComp
Banner None (Transform)
Speed 0.4
TestComp
Script TestComp
Banner Banner (Transform
Speed 7

UI

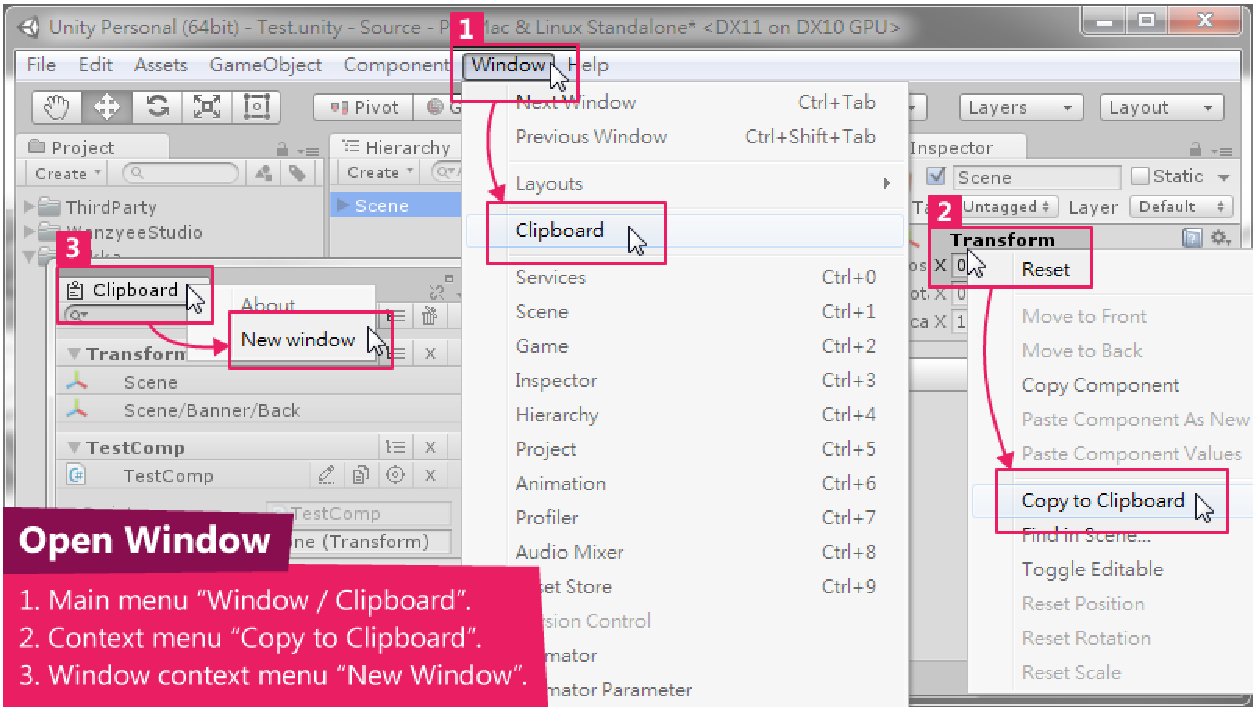
Simple, clear, and complete.
Easy to use and manage.

Overview:

- Easy to copy and paste multiple Components or Materials.
- Just drag'n'drop to snapshot them, and drag'n'drop to paste the copies back.
- All the copies can be managed and edited in a foldable Inspector-like window.
- With custom labels and quick search bar, it could work as a presets panel.
- One click to find the copy source.
- Most copies will be saved in the project.

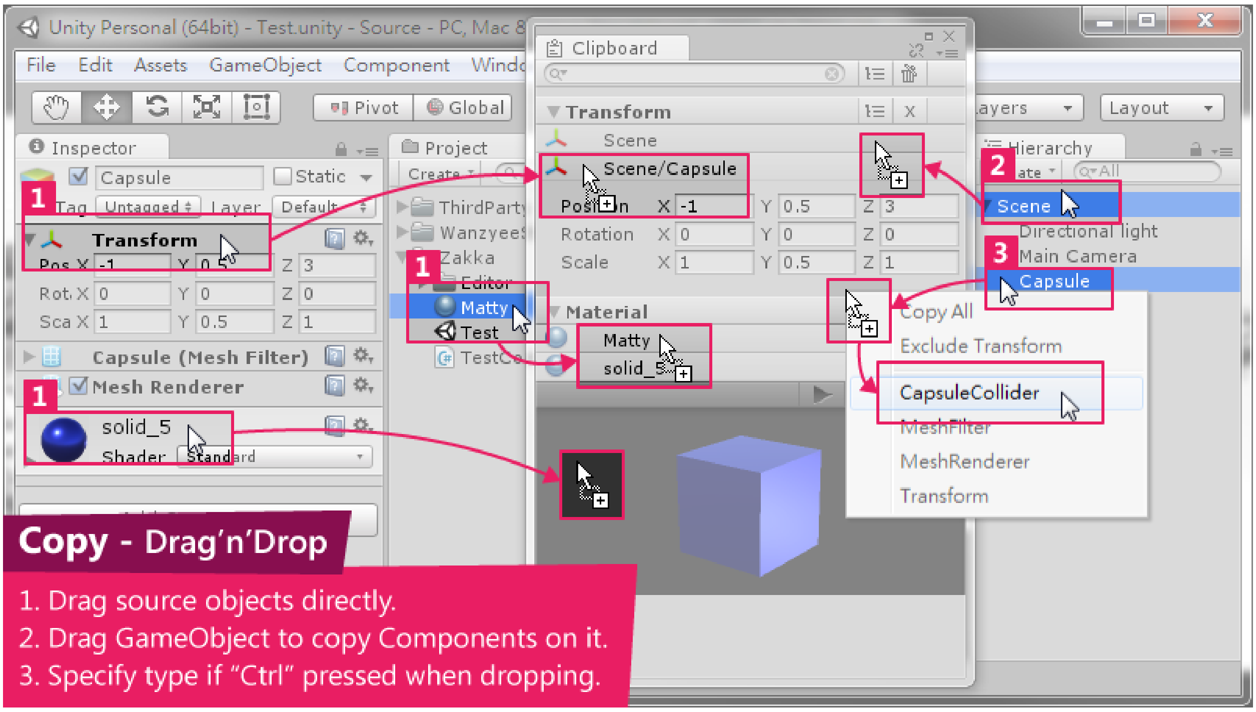
The further technical documentation is available here.
And the operation manual is right below.

Using:



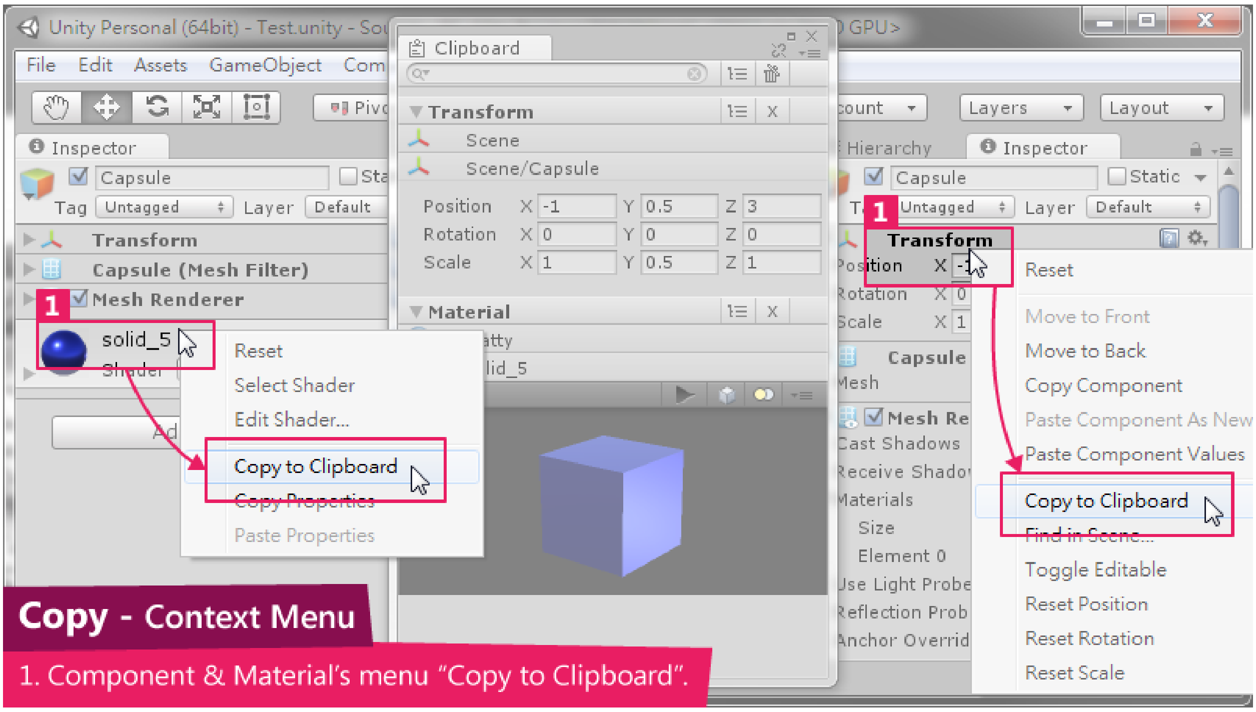
Open Window

1. Main menu “Window / Clipboard”.
2. Context menu “Copy to Clipboard”.
3. Window context menu “New Window”.



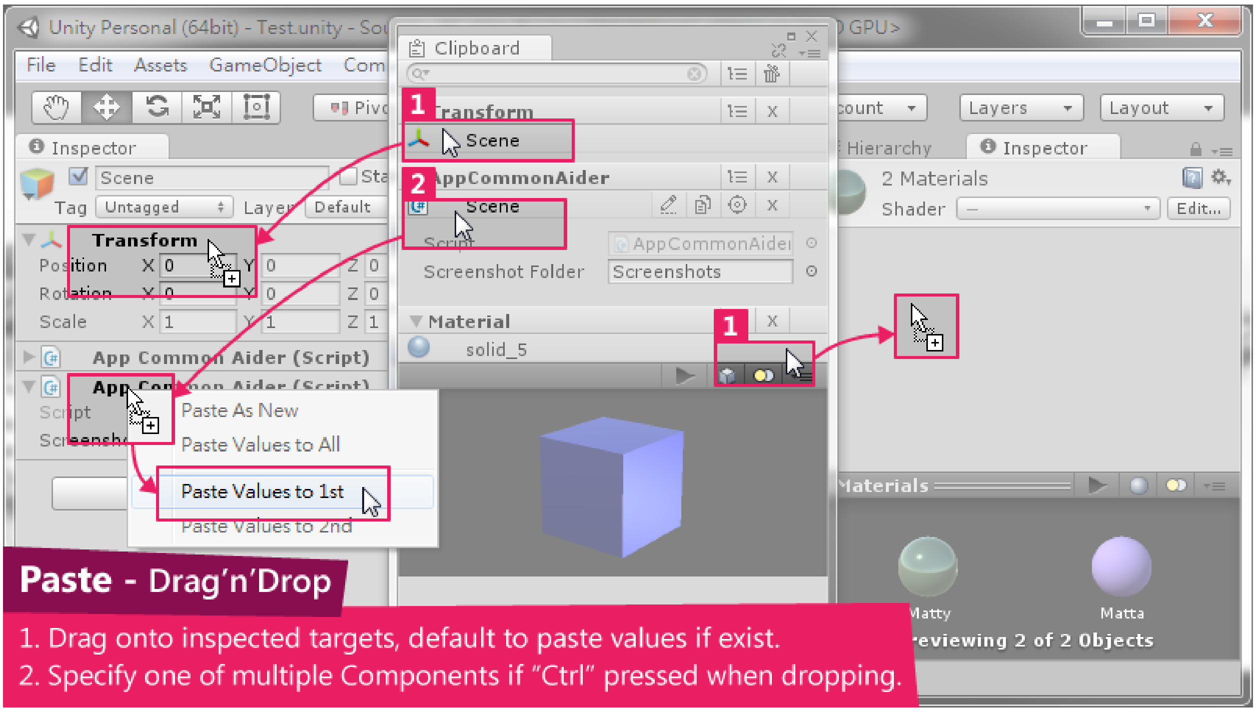
Copy - Drag'n'Drop

1. Drag source objects directly.
2. Drag GameObject to copy Components on it.
3. Specify type if "Ctrl" pressed when dropping.



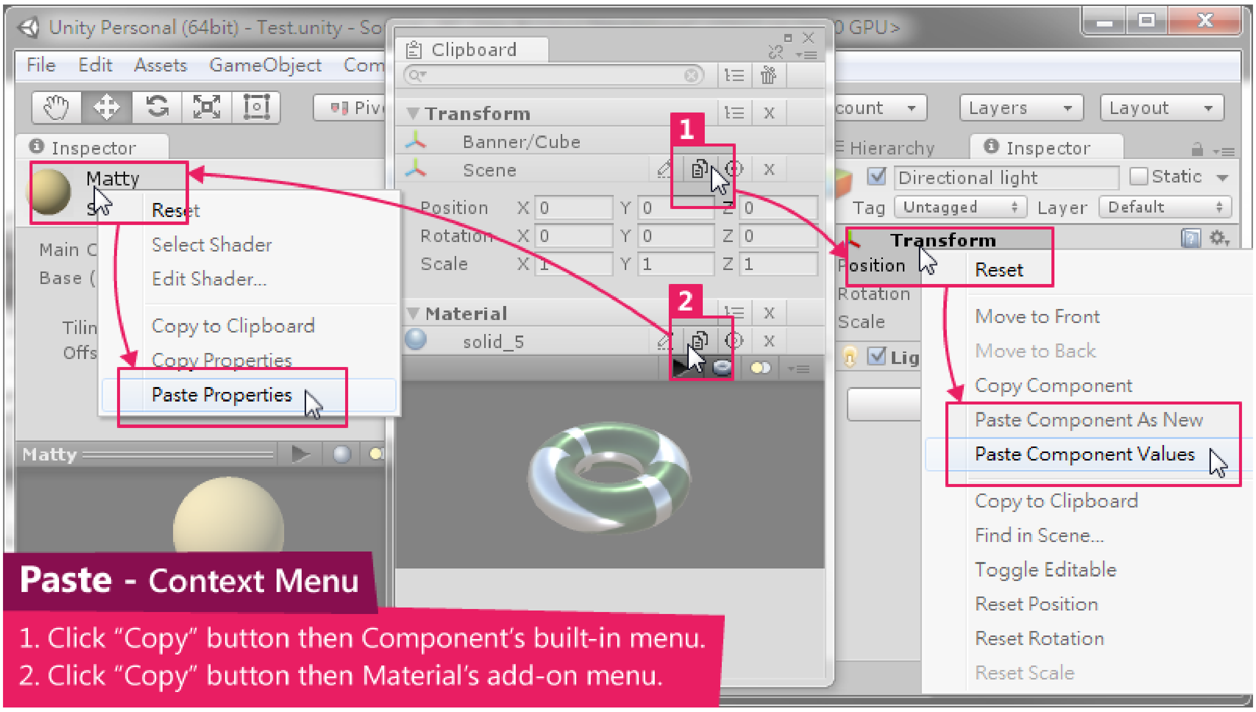
Copy - Context Menu

1. Component & Material's menu "Copy to Clipboard".



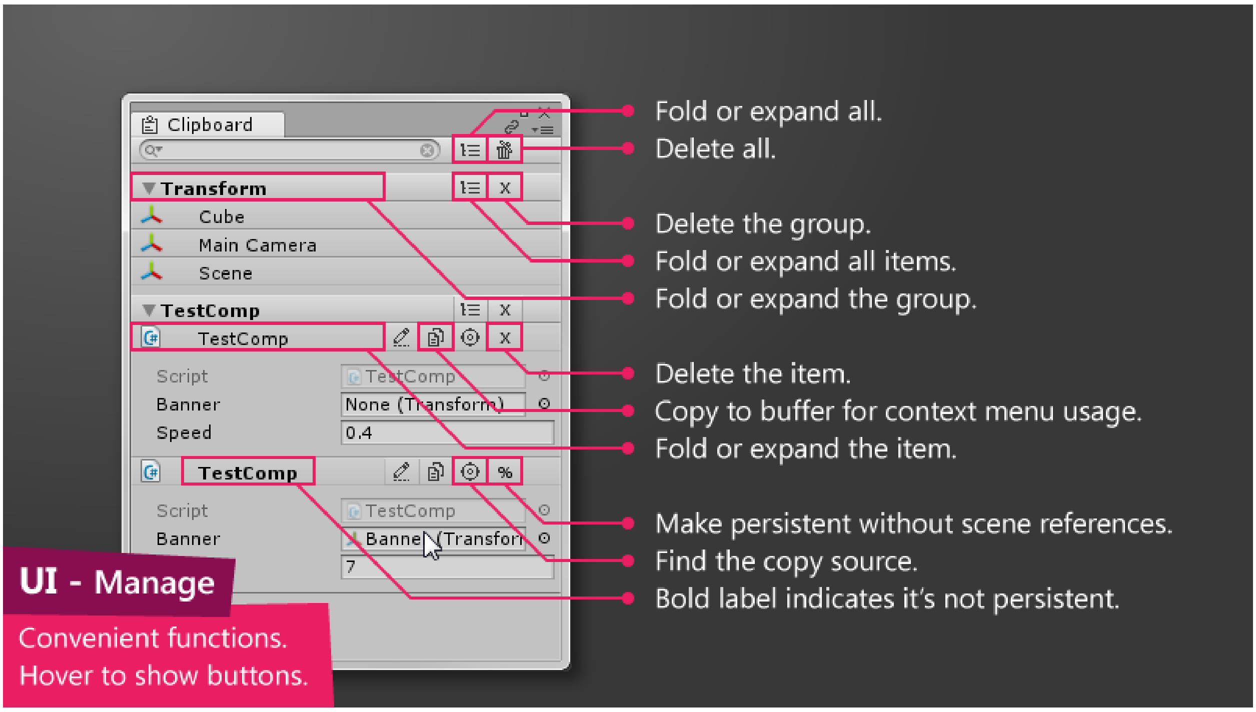
Paste - Drag'n'Drop

1. Drag onto inspected targets, default to paste values if exist.
2. Specify one of multiple Components if "Ctrl" pressed when dropping.



Paste - Context Menu

1. Click "Copy" button then Component's built-in menu.
2. Click "Copy" button then Material's add-on menu.



UI - Manage

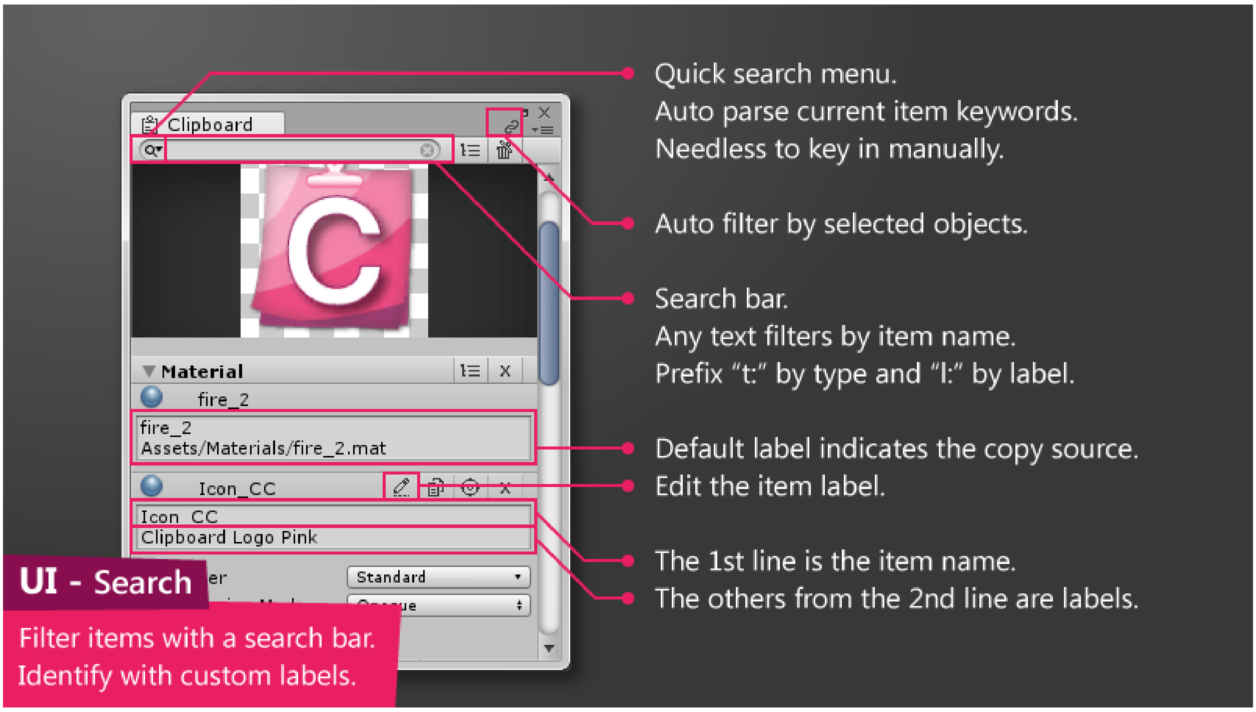
Convenient functions.
Hover to show buttons.

Fold or expand all.
Delete all.

Delete the group.
Fold or expand all items.
Fold or expand the group.

Delete the item.
Copy to buffer for context menu usage.
Fold or expand the item.

Make persistent without scene references.
Find the copy source.
Bold label indicates it's not persistent.



UI - Search

Filter items with a search bar.
Identify with custom labels.

Quick search menu.
Auto parse current item keywords.
Needless to key in manually.

Auto filter by selected objects.

Search bar.
Any text filters by item name.
Prefix "t:" by type and "l:" by label.

Default label indicates the copy source.
Edit the item label.

The 1st line is the item name.
The others from the 2nd line are labels.

Add-on:

- Here's **AuxCursor** to show cross windows dragging hint, useful for editor developer.
- This also includes a small utility: **Material Property Copier**.
- It provides context menus **"Copy Properties"** and **"Paste Properties"** for Material.
- It works just like the similar menus of a Component.

Release Notes:

1.2
- More features for free version, e.g., more copies, fold item, preview, find source, API.
- New, custom item label for search and identification.
- New, quick search menu to select filters without manual typing.
- New, auto filter by tracking selected objects.
- Change the search bar to work like the Project window's instead of regex.
- Keep the search filter after exiting the editor.
- Keep the fold states after exiting the editor.
- Clearer GUI, show item's buttons only when mouse hover, and adjust layout a little.
- Show a confirm dialog when click "Clear all" button to avoid miss operation.
- Save label data to AssetImporter.userData instead of a temp file in assets.
- Optimize, lower reload frequency with a new trigger system, obvious while playing.
- Optimize, new faster reload system for multiple windows usage.
- Change the store folder from "Temps/Clipboard" to "Temp/Clipboard".
- Change EditrixStyle API from "Create" to "Load", and reuse the generated texture.
- Fix, drag an object into a field of a copied item.
- Fix, Inspector reflection for dragging paste in Unity 5.5.0p3.
- Fix, support to copy audio filters.
- Improve compatibility for the same name types of different assemblies.

1.1.7
- Move the toolbar "Window" and "Tooltip" buttons to the window context menu.
- Change the toolbar "Clear" button icon instead of text.

1.1.6
- Fix the random crash caused by removing component in Unity 5.5.
- Fix the hideFlags error occurs when copying component in Unity 5.5.

1.1.5
- Add the documentation link to the window context menu "Help".
- Remove the big tooltip shows when there's no copied items.

1.1.4
- Change the drag'n'drop pasting system to avoid some weird behavior on Mac.
- It's able to cancel dragging paste by press "Esc" key.

1.1.3
- Fix showing the Save Assets dialog when copying, if the Verify Saving Assets flag set.
- Fix the null reference of an unexpected empty prefab when loading copies.
- Fix creating the style images from the window constructor.

1.1.2
- Fix an error when cleaning invalid Editor by checking its target.

1.1.1
- Add a regex search bar to filter labels.
- Rename the dll and move to an Editor folder to avoid exporting when building player.
- It's recommended to remove the old package before updating.

1.1
- New, Drag'n'Drop to paste.
- New, able to copy all components on a GameObject by dragging it.
- Fix copying a component which isn't a specific type, e.g., Halo is a Behaviour.

1.0.5
- Improve the source finding function to find in an unsaved scene.
- Fix the obsolete API about the HSV color used for icons.
- Fix the reference to the copy source when a copied component references to itself.
- Fix component copy operation to be not undoable since it shouldn't be.

1.0.4
- Fix the copied asset isn't saved caused by the previous bug fixing for weird object.
- Fix copying a material which inherits from Material in case.
- Fix the reference lost when a material refers to a non-asset texture.

1.0.3
- Add a dark background to every preview to make UI identical.
- Add a public method to open window, for users to make their own hotkey.
- Fix copying a component which inherits from Transform, e.g., RectTransform.
- Fix a weird object shown in Hierarchy when copy a component related to RectTransform.

1.0.2
- New, a button to easily find copy source object, and adjust the UI for it.
- Fix material copy operation to ensure be saved.

1.0.1
- Fix the cursor flashing when the Project or Hierarchy window changes constantly.
- Fix the component editor usage to avoid crash when a copied asset is operated manually.

1.0
- First release.

Publish Notes:

- Requires: Unity 5.3.5f1 or higher.
- Category: Editor Extensions / Utilities
- Keywords: Component Material Clipboard Copy Paste Snap Snapshot Drag Drop Editor Extension Utility Inspector Custom Preset Pool Search Filter

unity Asset Store Free

unity Asset Store Full

Full-only Features:

- Unlimited copy numbers. (10 for free version)
- Keep the copies after exit editor.
- Drag'n'Drop operations.
- Search filter operations.
- Custom item label.
- Open multiple windows.

Clipboard

Specific Classes » **Editor ExtensionsReleased Packages** » |

Component Clipboard

Utility to copy `UnityEngine.Component` or `UnityEngine.Material` and paste it back. More...

Inherits EditorWindow, and IHasCustomMenu.

Public Member Functions

void	**AddItemsToMenu** (GenericMenu menu) Add the window context menu items. More...

Static Public Member Functions

static void	**OpenWindow** () Show the clipboard window. More...
static bool	**IsCopyable** (Object source) Determine if the specified source is able to copy to clipboard. More...
static void	**Copy** (Object source, bool open=false) Copy the specified source to clipboard. More...
static void	**Clear** (Type type=null) Clear clipboard by specified type, or pass `null` to clear all. More...

Public Attributes

bool	**track** Flag to track selection to change search filter automatically. More...
string	**search** = "" The search filter pattern in the search bar. More...

Detailed Description

Utility to copy `UnityEngine.Component` or `UnityEngine.Material` and paste it back.

Useful for tweaking lots of objects in the editor, even in play mode. Click the menu "Window/Clipboard" to open the window. Just play and tweak, drag and drop, copy and paste whenever. It acts as sort of preset system, edit lots of copies as presets, and paste to apply values quickly. Easy to find and manage copies with regex search filter, custom item label, and foldable inspector.

To copy:

1. Drag'n'Drop from "Inspector" to copy the inspected sources.
2. Drag'n'Drop a `UnityEngine.GameObject` to copy the components on it.
3. Show menu to specify the component type if "Ctrl" pressed when dropping gameObjects.

4. Or click the context menu "Copy to Clipboard".

To paste:

1. Drag'n'Drop to "Inspector" to paste back to the inspected targets.
2. When dragging a component, it pastes values to the first one on the gameObject, or pastes as new if none.
3. Show menu to specify a target of multiple components if "Ctrl" pressed when dropping.
4. Or click the "Copy" button above any item, and paste by the target's context menu.

To filter items with the search bar:

1. Click the "Magnifier" icon to show a context menu made from the current items to select filters easily.
2. Or input any text to filter the item's name, just like the Project window's search bar.
3. Prefix "t:" filters by the type, the search will include all specified types.
4. Prefix "l:" filters by the tooltip as labels, an item has to match all specified labels.
5. Toggle the "Link" icon at the top-right of window to filter automatically by tracking selection.

To edit the item label:

1. Click the "Pen" button beside the label to show or hide the edit field.
2. To save the change, just press "Ctrl-Enter" keys or unfocus the field after editing.
3. The first line shows as the label title, and the full text is the tooltip.
4. Leave the field empty to show the default text, i.e., the copy source path.

To find the copy source object:

1. Click the "Aim" button to ping it or double-click to select it.
2. The saved trace path is also shown as the default label tooltip.
3. It'll beep if not found, e.g., the trace may be lost if the source is moved or renamed.

The reasons not to save the trace by references below:

1. A scene object reference will change when load a scene.
2. We can't save the scene reference in the project assets.
3. We shouldn't save the edit data in user's game scene.

For component references to scene object. This creates copies and store in editor scene temporarily to ensure content correct. It means the copy will be destroyed when quitting the editor. And also, the scene references will become missing when opening another scene. The situation above is applicable to a material with scene texture, too.

For material or component without reference to any scene object. The copies will be saved in an asset folder to make them still until "Clear" triggered manually. This tracks all copies by specific name or folder to ensure valid after script reloaded.

SVN users may clear all manually to avoid committing, or ignore the storing folder below:

1. The default is "Temp/Clipboard" under the root folder "Assets/WanzyeeStudio".
2. If the root is moved, it becomes "Temp/Clipboard" under the first found "WanzyeeStudio".
3. If there's no "WanzyeeStudio" folder, it'll be "Assets/Temp/Clipboard".

Instructions, to copy and paste generic component is dangerous, even if reflect all the fields. Since we'll never know what the developer do when the component awake. As the thread I commented, we'd be very careful with which aren't made by ourselves. Finally, this was created, works in the editor with Unity built-in classes and API.

Member Function Documentation

static void OpenWindow ( )

static

Show the clipboard window.

static bool IsCopyable (Object **source**)

static

Determine if the specified source is able to copy to clipboard.

Returns

`true` if is copyable; otherwise, `false`.

Parameters

source Source object.

static void Copy (Object **source**, bool **open** = `false`)

static

Copy the specified source to clipboard.

Optional to open window after copying.

Parameters

source Source object.

open If set to `true` open.

static void Clear (Type **type** = `null`)

static

Clear clipboard by specified type, or pass `null` to clear all.

void AddItemsToMenu (GenericMenu **menu**)

Add the window context menu items.

Menu "Help" to open the online manual, and "New window" to open another `Clipboard` window.

Parameters

menu Menu.

Member Data Documentation

bool track

Flag to track selection to change search filter automatically.

string search = ""

The search filter pattern in the search bar.

MaterialPropertyCopier

Specific Classes » Editor ExtensionsReleased Packages » |

Component Clipboard

Copy or paste properties from one `UnityEngine.Material` to another. More...

Static Public Member Functions

static void	**Copy** (Material source) Copy properties of specified material as source to paste later. More...
static bool	**Paste** (Material target) Paste properties of the source copied before to the specified target. More...

Detailed Description

Copy or paste properties from one `UnityEngine.Material` to another.

Operate by `UnityEngine.Material` context menu "Copy Properties" and "Paste Properties". This works like the similar menu of `UnityEngine.Component`, and will change the shader. To keep valid after script reloaded by a temporary material as medium storage. Copy properties by built-in `Material.CopyPropertiesFromMaterial()` method.

Member Function Documentation

static void Copy (Material source)

static

Copy properties of specified material as source to paste later.

Parameters

source Source.

static bool Paste (Material target)

static

Paste properties of the source copied before to the specified target.

Returns

`false`, if the source doesn't exist yet, otherwise `true`.

Parameters

target Target.

AuxCursor

Released Packages » **Component Clipboard**

Draw an auxiliary icon follows the cursor globally. More...

Inherits EditorWindow.

Properties

static DragAndDropVisualMode	**mode** `[get, set]` The visual mode of cursor to show, to set this will override the `icon`. More...
static Texture	**icon** `[get, set]` The texture of cursor to show, to set this will override the `mode`. More...
static Vector2	**offset** `[get, set]` The pixel offset between the auxiliary icon and the system cursor. More...
static Vector2	**size** `[get, set]` The pixel size of the auxiliary icon, set `Vector2.zero` to use the default texture size. More...

Detailed Description

Draw an auxiliary icon follows the cursor globally.

Used to provide a cross windows hint for reasons below:

1. `DragAndDrop.visualMode` only works when using `UnityEditor.DragAndDrop` system.
2. `EditorGUIUtility.AddCursorRect()` only applies inside the current window.

Property Documentation

DragAndDropVisualMode mode

static get set

The visual mode of cursor to show, to set this will override the `icon`.

The mode.

Texture icon

static get set

The texture of cursor to show, to set this will override the `mode`.

The icon.

Vector2 offset

static get set

The pixel offset between the auxiliary icon and the system cursor.

The offset.

Vector2 size

static get set

The pixel size of the auxiliary icon, set `Vector2.zero` to use the default texture size.

The size.

EditrixStyle

Released Packages » **Component Clipboard**

Extend editor GUI style, and include some premade icons or styles. More...

Static Public Member Functions

static Texture2D	**LoadTexture** (int width, Color32[] pixels) Load a `UnityEngine.Texture2D` with color `pixels` for editor usage. More...
static Texture2D	**LoadTexture** (string[] pixels, Dictionary< char, Color32 > colors, Color32 other) Trick to load a `UnityEngine.Texture2D` by parsing pixel `string` array. More...
static Texture2D	**LoadIcon** (string[] pixels, byte pro=196, byte free=60) Trick to load icon `UnityEngine.Texture2D` by parsing pixel `string` array. More...
static Texture2D	**LoadCursor** (string[] pixels) Trick to load cursor `UnityEngine.Texture2D` by parsing pixel `string` array. More...

Static Public Attributes

static readonly Texture2D	**splitterPixel** The splitter pixel texture with the same color as on the Inspector, 1 x 1. More...
static readonly GUIStyle	**statusBar** The status bar style. More...
static readonly Texture2D	**editIcon** The icon represents to edit, a pencil, 12 x 12. More...
static readonly Texture2D	**copyIcon** The icon represents to copy, overlapping notes, 12 x 12. More...
static readonly Texture2D	**deleteIcon** The icon represents to delete, a trash can, 12 x 12. More...
static readonly Texture2D	**aimIcon** The icon represents to aim, a front sight, 12 x 12. More...
static readonly Texture2D	**hierarchyIcon** The icon represents a hierarchy, indent level lines, 12 x 12. More...
static readonly Texture2D	**clipboardIcon** The icon represents a clipboard, 12 x 12. More...
static readonly Texture2D	**questionIcon** The icon represents a question mark, 12 x 12. More...
static readonly Texture2D	**linkIcon** The icon represents to link, a connected chain, 9 x 9. More...
static readonly Texture2D	**unlinkIcon** The icon represents to unlink, a broken chain, 9 x 9. More...
static readonly Texture2D	**stopCursor**

	The aux cursor represents a stop mark, 20 x 20. More...
static readonly Texture2D	**copyCursor** The aux cursor with a plus symbol, 19 x 15. More...
static readonly Texture2D	**linkCursor** The aux cursor with an arrow, 19 x 15. More...
static readonly Texture2D	**moveCursor** The aux cursor represents a dotted frame, 13 x 9. More...

Detailed Description

Extend editor GUI style, and include some premade icons or styles.

Member Function Documentation

static Texture2D LoadTexture (int width,
Color32[] pixels
)

static

Load a `UnityEngine.Texture2D` with color `pixels` for editor usage.

Return the texture with the same pixels created by this if exists, otherwise create new one.

Returns

The texture.

Parameters

width Width.

pixels Pixels.

```
static Texture2D LoadTexture ( string[]                    pixels,
                               Dictionary< char, Color32 > colors,
                               Color32                     other
                             )
```

static

Trick to load a `UnityEngine.Texture2D` by parsing pixel `string` array.

Array length as texture height, element `string` length as width. Set each `char` pixel by `colors` map if existing, otherwise the `other` color. Return the texture with the same pixels created by this if exists, otherwise create new one.

Returns

The texture.

Parameters

pixels	Pixel bits.
colors	Colors map.
other	Other.

```
static Texture2D LoadIcon ( string[] pixels,
                            byte     pro = 196,
                            byte     free = 60
                          )
```

static

Trick to load icon `UnityEngine.Texture2D` by parsing pixel `string` array.

Array length as icon height, element `string` length as width. Any space char as transparent, others color light if pro skin, otherwise dark. Return the texture with the same pixels created by this if exists, otherwise create new one.

Returns

The icon `UnityEngine.Texture2D`.

Parameters

pixels	Pixel bits.
pro	Grayscale for pro skin.
free	Grayscale for free skin.

static Texture2D LoadCursor (string[] pixels) static

Trick to load cursor `UnityEngine.Texture2D` by parsing pixel `string` array.

Array length as cursor height, element `string` length as width. Any '1' char as white, '0' as black, otherwise transparent. Return the texture with the same pixels created by this if exists, otherwise create new one.

Returns

The cursor `UnityEngine.Texture2D`.

Parameters

pixels Pixel bits.

Member Data Documentation

readonly Texture2D splitterPixel static

The splitter pixel texture with the same color as on the Inspector, 1 x 1.

readonly GUIStyle statusBar static

The status bar style.

Text colored light if pro skin, otherwise dark, with darken background anyway.

readonly Texture2D editIcon static

The icon represents to edit, a pencil, 12 x 12.

readonly Texture2D copyIcon static

The icon represents to copy, overlapping notes, 12 x 12.

readonly Texture2D deleteIcon static

The icon represents to delete, a trash can, 12 x 12.

readonly Texture2D aimIcon

static

The icon represents to aim, a front sight, 12 x 12.

readonly Texture2D hierarchyIcon

static

The icon represents a hierarchy, indent level lines, 12 x 12.

readonly Texture2D clipboardIcon

static

The icon represents a clipboard, 12 x 12.

readonly Texture2D questionIcon

static

The icon represents a question mark, 12 x 12.

readonly Texture2D linkIcon

static

The icon represents to link, a connected chain, 9 x 9.

readonly Texture2D unlinkIcon

static

The icon represents to unlink, a broken chain, 9 x 9.

readonly Texture2D stopCursor

static

The aux cursor represents a stop mark, 20 x 20.

readonly Texture2D copyCursor

static

The aux cursor with a plus symbol, 19 x 15.

readonly Texture2D linkCursor

static

The aux cursor with an arrow, 19 x 15.

readonly Texture2D moveCursor

static

The aux cursor represents a dotted frame, 13 x 9.