

uzLib.Lite

The Lite version of my paid uzLib (Unity3D) API used as dependency of my open-source repos.

See full PDF at: <https://dev.z3nth10n.net/dev/assets/getassetpreview/manuals/README.pdf>

Contents

This API has an extensive list of methods to extend functionality of .NET Framework classes, WinForms, System.Console and Unity3D, also it contains own/third party implementation:

List of interesting contents by namespaces

- **uzLib.Lite.Core**

- Contains a Singleton implementation.
- Contains a Console wrapper to read lines or keys at the same time. (**ConsoleOutput** class, *method: ReadLineOrKey*)
- Inside of the Input namespace we can find some class that contains an implementation (**SmartPsswr** class) to read passwords in screen without displaying it (as expected).

- **uzLib.Lite.Extensions**

- Contains some *array/collection extensions* (add element to an array, check if *Dictionary/List/HashSet/IEnumerable* is null or empty (*IsNullOrEmpty*), *ForEach* extensions to execute *Action* callbacks (included with index) (expanded on paid API), remove at index, *DisctintBy...*)
- Contains some Dictionary extensions methods:
 - **FindIndex** (find item index by predicate)
 - **GetIndex** (get index of existing item, if not, returns -1)
 - **AddOrSet** (add value if key doesn't exists, if not update its value)
 - **AddOrAppend** (the same as before, but for *List<T>/T[]* as *TValue*)
 - **AddAndGet** (if key doesn't exist add it with specified value, then return it)
 - **Get** (the same as before, but force generic types with constraints to be classes (where *TValue : class, new()*))
 - **SafeGet** (try to get value if key exists if not returns default value (*default(TValue)*))
- Contains some *Bitmap extensions*: get an *IEnumerable* of Colors from a *Bitmap* class or *Save* it to a path (or to a Stream (**TODO**)).
- Contains some *byte extensions*: *RoundOff* method (to get a rounded byte)...
- Contains some *Color extensions*: **get threshold** between two colors, compare *color1* to **get similar color** from a *IEnumerable* of colors (2), *posterize convolution*, get percentage of similarity, round color...
- Contains a method extension to **compile *.sln files...**
- Contains *compression extensions*: **zip/unzip** files from it's path, streams or objects (sync/async)...
- Contains a *CodeDOM extension* to add *static* literal to new classes declared through *CodeTypeDeclaration* class.
- Contains a *DateTime extension* to get its *UNIX timestamp*(in *DateTime* not *DateTimeOffset* class).
- Contains a *Exception extension* to check if *array/List/Dictionary* specified index is **out of bounds**.
- Contains *Git extensions* to get some data by using *LibGit2Sharp* library.
- Contains *HTML extensions* to **clean and find ocurrences** on *HTML source code*.
- Contains *IO exntensions* to check valid paths, get file names from urls, get and check relatives paths patterns, check if path is a directory, get top level directories, go up in the tree (*N times*), get temporaly directories, check if directories are empty or null, delete folder contents...
- Contains *Net extensions* to download files and make (*GET*) requests...
- Contains a *Object extension* to check if object is casteable...
- Contains a *Process extension* to run process asynchronously...
- Contains *Reflection extensions* to invoke static/non-static methods, check if method exists in Assembly, run method by checking exceptions...
- Contains *Serialization methods* to serialize/deserialize files/Streams/objects/byte arrays and to check if valid JSON/XML, etc...
- Contains *String extensions* to check if strig is null or empty (no longer needed to use *...string.IsNullOrEmpty("example") ...*), format strings (the same logic), to change the first character from a string in uppercase...
- Contains a *URI extension* to check if URL passed by string is valid.
- Contains a *Visual Studio extension* to get the Startup project from a solution file...

- **uzLib.Lite.Plugins.SymLinker**

- Contains a system to create **symbolic links** in Windows/Linux/Mac OSX...

- **uzLib.Lite.Shells**

- Contains an **obsolete** `GitShell` (an self-implementation of Git functionality before I discovered `LibGit2Sharp`...)

- **uzLib.Lite.Unity.Extensions**

- Contains *Animations extensions*.
- Contains *Color extensions*: Orange color (**TODO**: implement more colors), check if colors are similar, the distance between colors, get random colors, the same extensions from `uzLib.Lite.Extensions.ColorHelper` class (but adapted for Unity3D)...
- Contains *Geometry extensions* to get random positions (`Vector3`), get average of `Vector3s`, get encapsuled bounds (this is an average of the bounds of all renderers of an `GameObject`s), getting the orthographic size to set in a camera to make this object fit perfectly on its view, get the offset from a model (getting the maximum and the minimum centers from the `Renderers` of a `GameObject`s), compare *Vector3/Vector2* types to check if lesser or equal, greater or equal, lesser/greater than another *Vector3/Vector2*, get inverted `Vector3/Vector2`, get the ray from a camera center...
- Contains *Math extensions* get the maximum absolute of an array of floats (`params`), getting the multiple of a float, check if float is between a range (exclusive/inclusive operation), set the "Y" parameter of a vector, get the distance from two `Vector3/Vector2`...
- Contains *Object extensions (GameObject/Transforms)* to set layers, tags recursively, remove components recursively, find Transform parent by its name, get the topmost parent component, the components by name, get or add component in a `GameObject` (or throw exception/`LogError`), make child easily, safe destroy (check if `Application.isEditor` to execute `DesroyImmediate` instead), get `GameObject` path (dump hierarchy of childs into a string), send message to objects of selected type...
- Contains *Rect extensions* to make `PropertyDrawers` implemenentation easier, check if rect is inside another rect, add padding to a rect, Clamp them...
- Contains *Texture extensions* to write a text into a texture, get its width/height as a `Vector2`, create a texture from a color...
- Contains *UI extensions* where are located some custom `GUIStyles`, get centered `Rect`, draw a `Rect`, draw a `ProgressBar`, centered label, draw `Marquees`...

- **uzLib.Lite.Unity.Utils**

- Contains a *GLDebug class* to visualize Gizmos in-game.

Setup

Need help? Just clone [this repository](#) into your Unity project or in your solution.

For documentation, just check the [Documentation API](#).

Issues

Having issues? Just report in [the issue section](#). Thanks for the feedback!

Contribute

Fork [this repository](#), make your changes and then issue a pull request. If you find bugs or have new ideas that you do not want to implement yourself, file a bug report.

TODO

- There are some methods missing from the text above.
- Change `Safe` word on methods and use `Try`.

Donate

Become a patron, by simply clicking on this button (very appreciated!):



... Or if you prefer a one-time donation:

[Donate](#)



Copyright

Copyright (c) 2019 z3nth10n (United Teamwork Association).

License: GNU General Public License v3.0