CLIU - Command Line Interface for Unity

Lucas Sampaio Dias
lucassampaiodias@gmail.com

Introduction

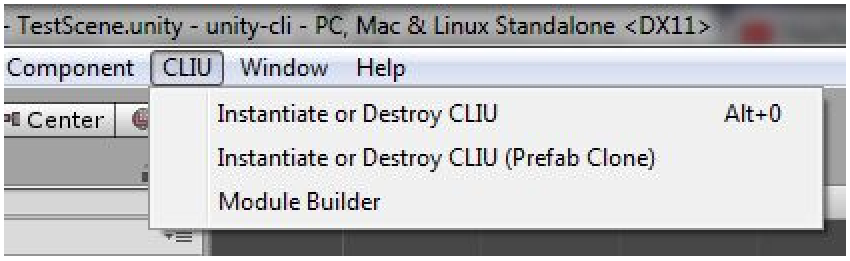
Welcome to the CLIU documentation! In this document you should be able to find all the information you need to use CLIU on your project, as well as many useful tips on how to make the most of this plugin.

If you need to customize anything, all UI elements are made using Unity's default UI solutions (NGUI), so feel free to use what you already are familiar with to adapt the CLIU window. Also, the source code for CLIU is available if you want to change or add anything.

Finally, be sure to send me an email (lucassampaiodias@gmail.com) if you have any suggestions of useful features or bugs to report.

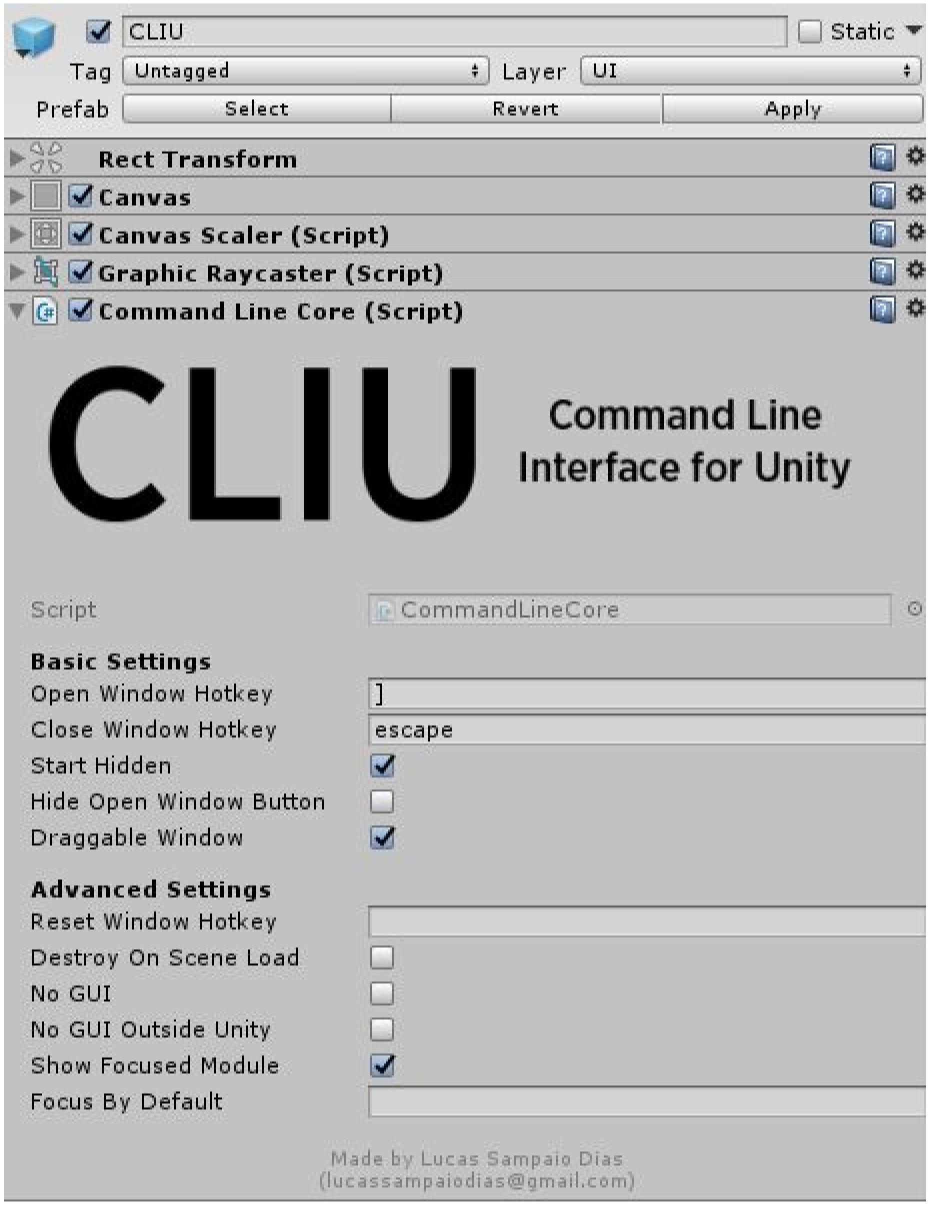
Getting Started

Integrating CLIU to your project is really simple! Go to the ‘CLIU’ submenu and click on ‘Instantiate or Destroy CLIU’, or just press Alt + 0.


- TestScene.unity - unity-cli - PC, Mac & Linux Standalone <DX11>
Component
CLIU
Window
Help
Center
Instantiate or Destroy CLIU
Alt+0
Instantiate or Destroy CLIU (Prefab Clone)
Module Builder

I suggest you to add the CLIU prefab on the first scene that is loaded when your game launches. By doing this, CLIU will always be available by default when you play the game. However, if you want to add CLIU to specific scenes, do the process above for each scene and change the ‘Destroy on Scene Load’ setting of the prefab to true (you can find the prefab at CommandLine/Resources).

And that’s it! After placing the CLIU prefab into your scene everything should work already. Press Play to test your game and the CLIU window should appear if you press the ‘Open CLIU’ button or press ‘]’ on your keyboard.


CLIU ☐ Static
Tag Untagged Layer UI
Prefab Select Revert Apply

Rect Transform
✓ Canvas
✓ Canvas Scaler (Script)
✓ Graphic Raycaster (Script)
✓ Command Line Core (Script)

CLIU Command Line Interface for Unity

Script CommandLineCore

Basic Settings

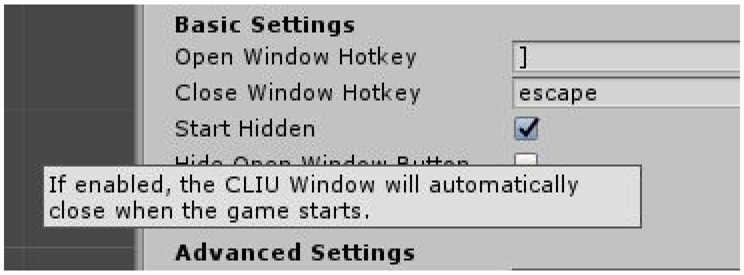
Open Window Hotkey	]
Close Window Hotkey	escape
Start Hidden	✓
Hide Open Window Button	☐
Draggable Window	✓

Advanced Settings

Reset Window Hotkey	
Destroy On Scene Load	☐
No GUI	☐
No GUI Outside Unity	☐
Show Focused Module	✓
Focus By Default	

Made by Lucas Sampaio Dias
(lucassampaiodias@gmail.com)

Be sure to take a look on all the available settings to make the most of CLIU. There is a tooltip for all settings if you need any help.


Basic Settings

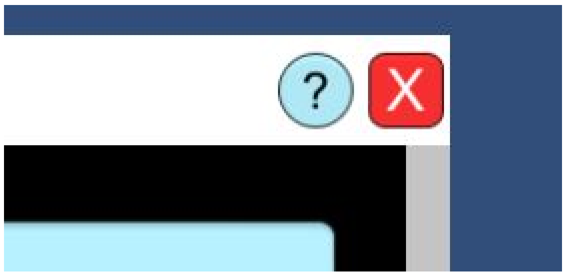
Open Window Hotkey	]
Close Window Hotkey	escape
Start Hidden	✓

If enabled, the CLIU Window will automatically close when the game starts.

Advanced Settings

Commands and Modules

Using CLIU should feel quite similar to using pretty much any other CLI. All you have to do is open the window, type your commands (separated by spaces) and press Enter/Return after typing the entire command. To see some basic commands hover over the ‘?’ next to the close button.



Commands are separated into two types: ‘Core’ commands and ‘Module’ commands.

Core commands are read first and have priority in execution. Below is a list of all Core commands:

- help (which can also be called by ‘h’ and ‘-h’)
- modules (which can also be called by ‘m’)
- hide (which can also be called by ‘close’)
- exit
- clear
- reset
- focus

Module commands are “custom” commands built on top of CLIU. These are what makes a command line interface useful. There are many modules that come with CLIU by default (like the ‘transform’ module, that comes with commands that manipulate the transform attributes of Game

Objects), but you can also make your own modules to satisfy the needs of your projects. More on that on the next section of this document.

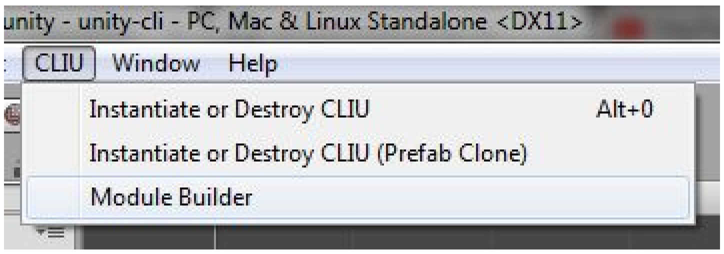
Below is a list of *some* of the awesome commands found on the default modules:

1. time timescale 1: sets the timescale to the value specified
2. time slowmo 0.5 3: sets the timescale to 0.5 for 3 seconds
3. scene loadscene 5: load the scene with build index equals to 5
4. obj call Player Die: call the method Die() on the Player game object
5. obj destroytag Enemy: destroy all game objects the the Enemy tag

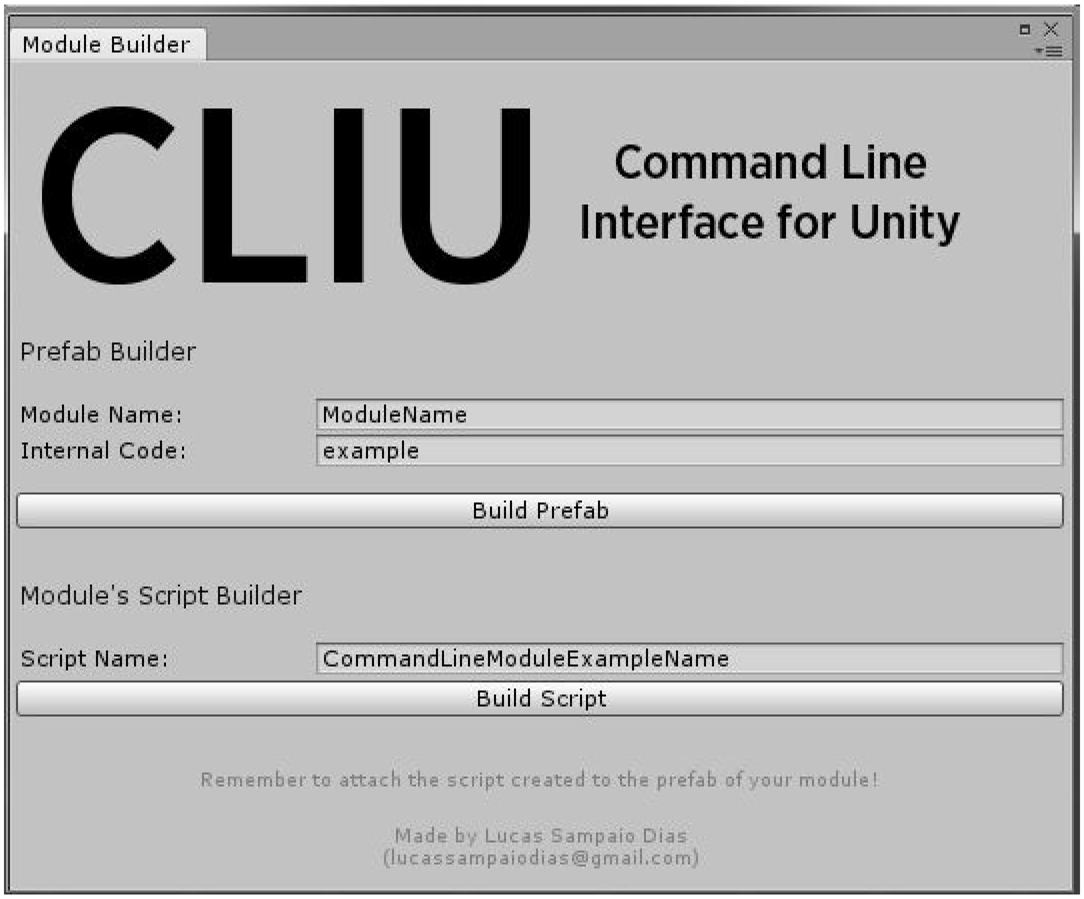
Two **very** important things you should know. The first thing is that the first word on the commands above are the module codes of where these commands are implemented. The second thing is that **you don't have to use the module code** every time you want to use a module command! For the first command, just use "timescale 1" instead and it will work just fine! The only "side effect" is that all modules with that same command will be executed. It is worth noting that all commands available on the default modules are unique, so this problem of two or more commands being executed will never happen for them.

Creating Custom Modules

Building new modules is made very easy using the Module Builder that comes with CLIU. To open it, go to the CLIU submenu and click on ‘Module Builder’.


unity - unity-cli - PC, Mac & Linux Standalone <DX11>
CLIU
Window
Help
Instantiate or Destroy CLIU
Alt+0
Instantiate or Destroy CLIU (Prefab Clone)
Module Builder

This should open a window just like the one on the image below:



Prefab Builder

Module Name: ModuleName

Internal Code: example

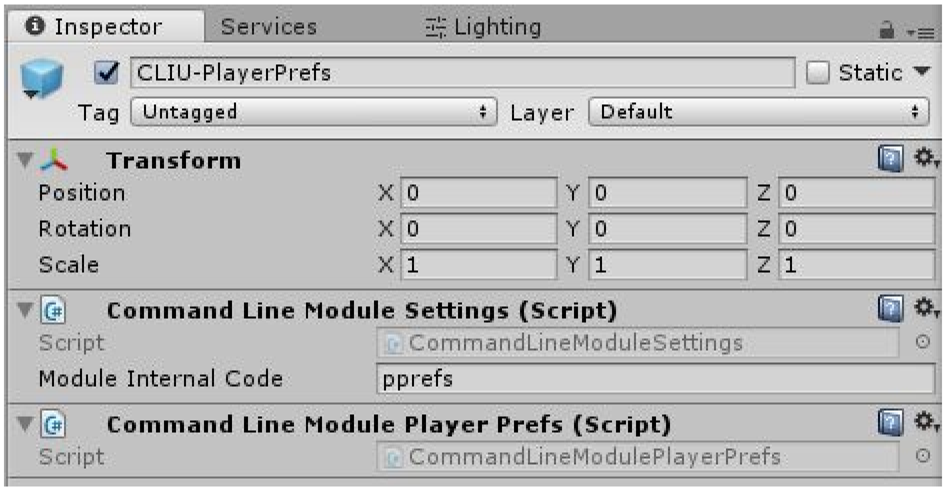
Module's Script Builder

Script Name: CommandLineModuleExampleName

Remember to attach the script created to the prefab of your module!

Made by Lucas Sampaio Dias
(lucassampaiodias@gmail.com)

First, insert the name of the module (which will be the name of the prefab) and the module's code (like pprefs is the internal code for the PlayerPrefs module, for example) and press 'Build Prefab'. Then, insert the script name of your module (I suggest CommandLineModulePlayer for a module called Player, for example) and press 'Build Script'. Finally, find the module prefab (CommandLine/Resources/Modules) and attach to it the script you created. The prefab should look like this:



CLIU-PlayerPrefs ✓
Static
Tag: Untagged
Layer: Default

Transform

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

Command Line Module Settings (Script)
Script: CommandLineModuleSettings
Module Internal Code: pprefs

Command Line Module Player Prefs (Script)
Script: CommandLineModulePlayerPrefs

When you press Play again the module will be already working with a few example commands. To add the commands you want open the script the Module Builder created (the one attached to the module prefab) and change the code of the Execute() method. If you need any guidance, follow the patterns used on the scripts of the default modules.