CMD+ Documentation

Console Commands

To create your own console commands, simple create a **public static** method anywhere in your codebase. This method may return any value you wish and will print its value using ToString(). You can add as many parameters to the command as you wish. They are automatically casted if they are of one of the following types:

- Int
- Float
- Bool
- String
- Color (given as string like so: r/g/b/a   example: 1/0/0/1 = full red)

To mark the method as a command, simply add the CMDCommand attribute to the method. The attribute has 2 parameters. The first one is the description, this will show up next to the command when the List command is called. The second parameter is a boolean which defaults to false. If set to true, the command will be hidden when the List command is called. Here is an example of a valid command:

```
[CMDCommand("A description of the command")]
public static void CommandWithDescription() {
    Debug.Log("Executed a command with a description!");
}
```

Console buttons

Buttons on the right side of the console can be made similarly. Simple add the attribute CMDCommandButton to the **public static** method. This method also needs to have **void** as its return time and must have **no parameters**. You can add both the CMDCommand and the CMDCommandButton attributes to the same method to make it both a command, as well as a button. The attribute has 2 parameters, the first one is the text to be displayed on the button. The second is a string that will be casted to a color for the button's color. The format for this is; R/G/B/A. The alpha channel is optional however. Here is an example of a valid CMDCommandButton attribute that creates a red button:

```
[CMDCommandButton ("Red Button", "1/0/0/1")]
```

CMDSettings

CMD+ comes with its own settings system. These are primitive values stored in XML values outside of your game's build. This allows you to edit the values without having to rebuild the game. The xml files are automatically included with your build. Please note however that this feature is not supported on mobile. There are however a few UI scaling settings for mobile that are compiled with the game. The XML files are located in *[Path_to_root]/CMD/Settings*. You can either edit the values of the settings by editing the xml files directly. Or you can use the editor window which you can open by clicking **Tools -> CMD+ -> Settings**. All types serializable through XML are supported, however the editor only supports the following types:

- Int
- Float
- Bool
- String
- Color (given as string like so: r/g/b/a   example: 1/0/0/1 = full red)

Setting explanation

ATTACH_TO_UNITY_CONSOLE: If set, all messages printed in the unity console are automatically sent to the CMD+ console as well.
CMDPLUS_ENABLED: This will toggle the scripting symbol *CMDPLUS_EXCLUDED_IN_BUILD*. When set, all functionality of CMD+ are disabled. You can still call the methods, so you will not need to change your code accordingly.
CONSOLE_OPEN_ON_ERROR: When set to true, the console will automatically open once an error is thrown.
TOGLLE_KEYCODE: The keycode used to toggle the console on and off on PC, Mac & Linux. If set to [DEFAULT], the default key values are used instead. These are *F1* on Windows & Linux. And *Tab* on Mac.

CMDLogger

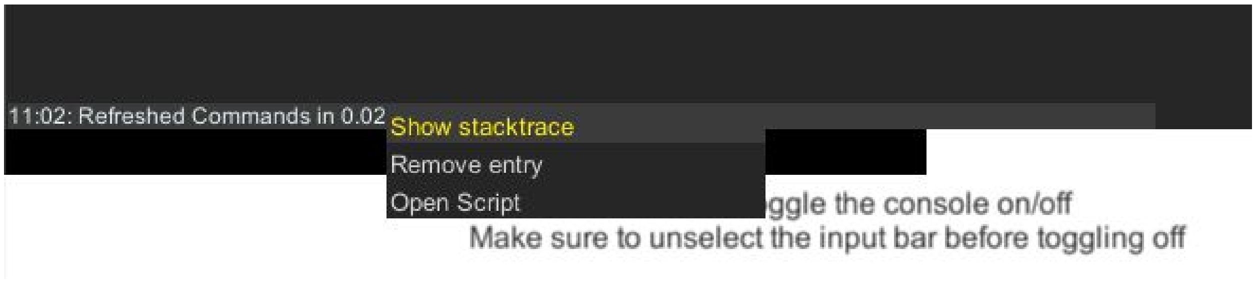
If the CMDSetting *SAVE_LOGS_ON_EXIT* is enabled. A .txt file with all log messages is saved in the *[Path_to_root]/CMD/Logs* folder. You can limit the amount of messages with the *LOG_MAX_AMOUNT* setting. Please note however that this feature is not supported on mobile.

Mobile

CMD+ also works on mobile devices. The CMDSettings and CMDLogger features are not supported on mobile however. To turn the console on on mobile, simply tap the upper left and right corners of the screen multiple times. To close it, tap the lower left and right corners. You can edit the CMDMobileSettings asset to change the UI scaling to better suit your needs.

Context menu

You can view the stacktrace of log entries by right clicking on them and selecting *Show stacktrace*.
The stacktrace view can be moved around by dragging it with the mouse.



```
11:02: Refreshed Commands in 0.02
ggle the console on/off
Make sure to unselect the input bar before toggling off
```

You can remove entries by click *Remove entry*.
You can also open the script if the highest stack frame by clicking *Open Script*. This will open the file through Unity's AssetDatabase class.

CMDLogEntry

When calling CMD.Log("your message"), you will be returned the CMDLogEntry of that message. You can alter the content of this message using the Message property. This can be useful if you want to live update the message with something.

For any further questions, please contact me using either my contact form here: http://daanruiter.net/contact.php or email me directly to contact@daanruiter.net.