

Methodical V1.2.3

A C# code console for Unity

Introduction

Thank you for purchasing Methodical!

Methodical is an editor extension for Unity that provides a command-line interface for any MonoBehaviour attached to an object and for static classes. Methodical allows the user to access variables and call methods on MonoBehaviors or static classes in a quick, painless, and familiar c#-like syntax.

Table of Contents

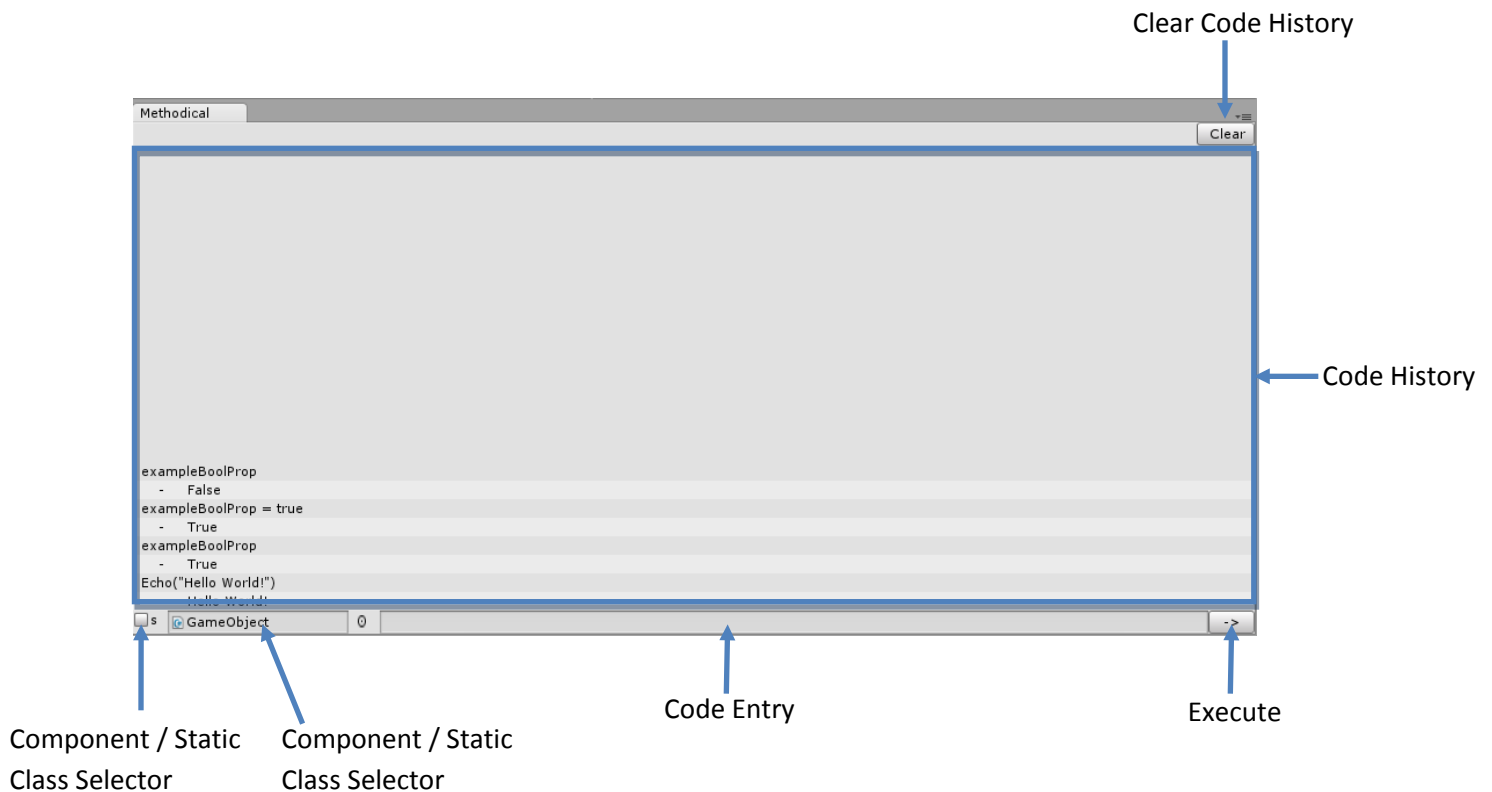
Introduction	1
Table of Contents	1
Starting Methodical	2
Getting to Know Methodical.....	2
Using Methodical in Instance Mode	2
Using Methodical in Static Mode.....	4
Conclusion.....	5
Change log.....	6
Limitations	6
Version 1.2.3 Known issues	6

Starting Methodical

Methodical can be found in the Unity menu bar under *Window -> Methodical*.

Getting to Know Methodical

Here are the different parts of the Methodical window:



Using Methodical in Instance Mode

Instance mode works on components attached to game objects in your scene or hierarchy list.

1. Start Methodical.
2. Make sure the *Component / Static Class Selector* is unchecked (located in the bottom left corner of the Methodical window)
3. Drag a script from the inspector to the *Component Selector* in the bottom left corner of the Methodical window to start.

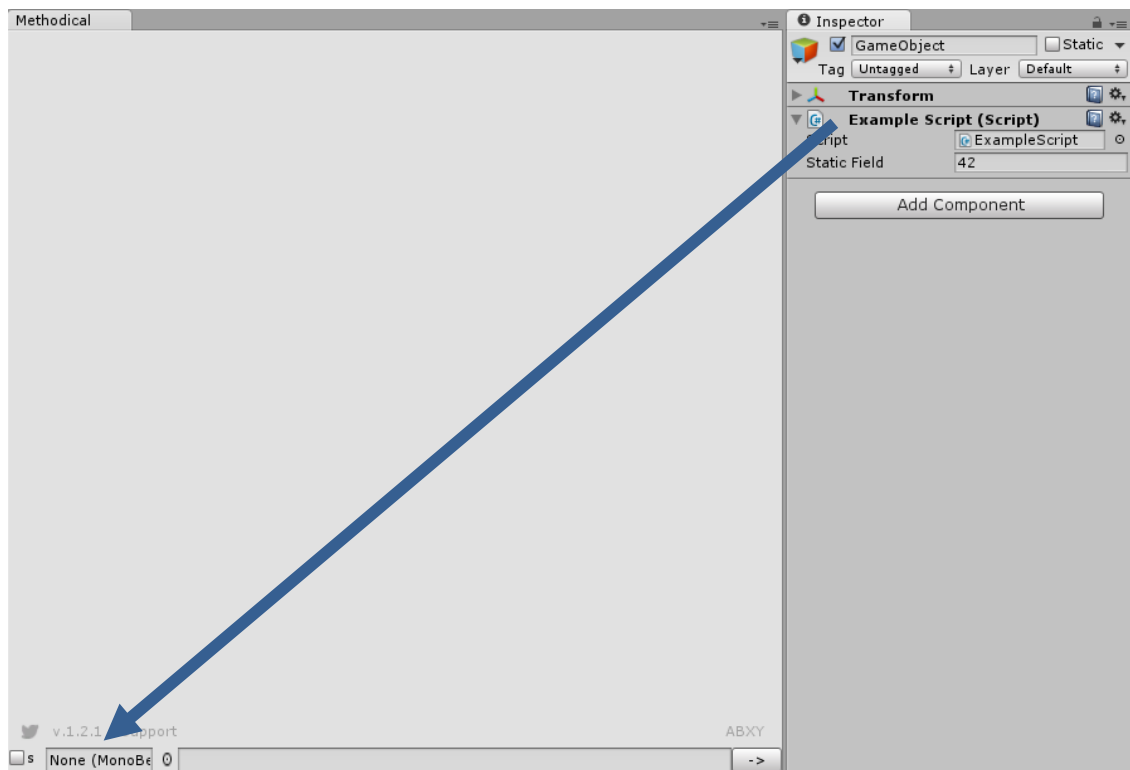


Figure 1: Drag a script to the object picker to start!

4. Write code statements in the text field next to the object picker and press enter or the *Execute* button to submit. Methodical accepts code statements written in c# in the context of the MonoBehaviour Methodical is operating on. For example, consider a class called “ExampleScript”, shown below:

```
public class ExampleScript{  
    public float foo = 42;  
    public string Echo(string message){  
        return message;  
    }  
}
```

In order to retrieve the value of foo the user can type the variable name “foo” into the command line (no semicolon necessary). Methodical will print the value of the variable to the window. Likewise, the user can call the Echo method by typing “Echo(“give me this string”)” which will print the return value of the Echo method to the Methodical Window.

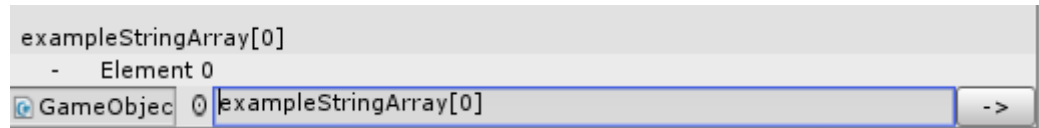


Figure 2: Retrieving an array's value at index 0

Methodical also features a code history. The user can recall previously used commands using the up and down arrow keys.

Using Methodical in Static Mode

Static mode functions exactly the same as instance mode, except it operates on static classes, methods, and fields rather than the members of components attached to gameobjects.

1. Start Methodical
2. Make sure the Component / Static Class Selector is checked (located in the bottom left corner of the Methodical window)
3. Type the name of the static class you wish to operate on in the Static Class Selector, as shown below:

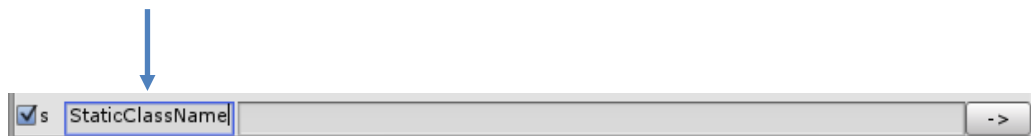


Figure 3: Enter your static class name here

4. Use Methodical as you would in instance mode (see the previous section, step 4), except on the static members of the class.

Conclusion

We hope that you enjoy using Methodical! If you have any questions please contact us!

abxygames.com

https://twitter.com/mark_at_ABXY

mark@abxygames.com

Report a bug:

Flyswatter.abxygames.com

Other Useful Info

Change log

- 1.2.3
 - Fixed a bug where floats were not being parsed properly when decimals were included
- 1.2.1
 - Added assignment to variables! Assign to variables just like you would in C#
 - Added support for running Methodical on static classes.
 - We changed our name to ABXY! Changed branding to reflect this.
- 1.1.1
 - Initial release!

Limitations

- Methodical only works with single line statements by design. Because of this, code flow statements like if, else, etc. are not supported and will not likely be supported in future versions.
- Methodical will break if moved. Install in the assets folder only and leave it there. Future versions will not have this limitation.

Version 1.2.3 Known issues

1. Methodical fails gracefully in the event that user input has bad syntax or references nonexistent objects. That said, there are some edge cases left that will post errors to the debug console, particularly with mismatched types for dictionary, list, and array indexes. We are working hard to eliminate these errors, but in the meantime this shouldn't affect normal operation significantly.