



Methodical V1.1.1

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. Methodical allows the user to access variables and call methods on a MonoBehaviour in a quick, painless, and familiar manner using a c#-like syntax.

Using Methodical

1. Start Methodical. Methodical can be found in the Unity menu at Window->Methodical.
2. In order to use Methodical, the user must first specify a MonoBehaviour to operate on. Drag a script from the inspector to the object selector in the bottom left corner of the Methodical window.

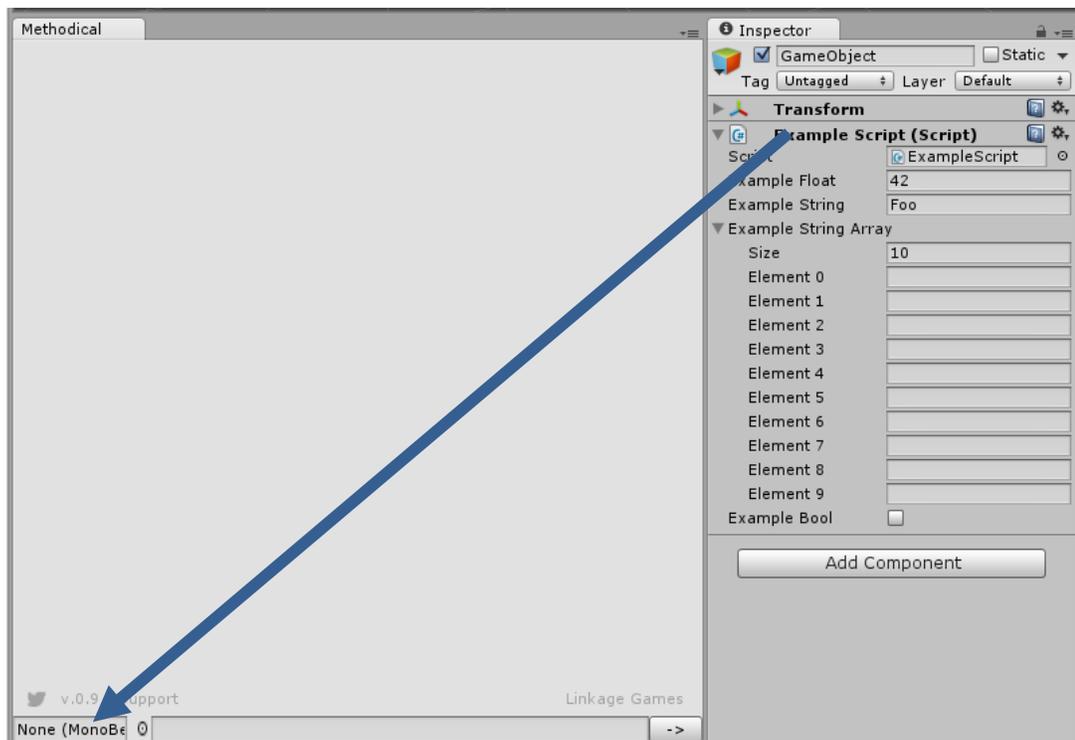


Figure 1: Drag a script to the object picker to start!

3. Write code statements in the text field next to the object picker and press enter or the arrow 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.

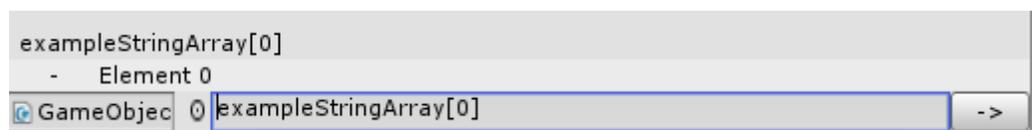


Figure 2: Retrieving an array's value at index 0

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

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.

- Currently Methodical does not support assignment operations using the “=” operator. This will be changing in the very near future, but in the mean time setter methods work fine!
- Methodical supports retrieval of nearly any data type from a class, including custom data types. However, Methodical does not support parsing all data types as method arguments. We have added support for the types users use most often and most shouldn't notice any problems. We are rapidly working to increase the number of supported types. Notable missing types include Vector2, Vector3, and Double, which are coming very soon.
- Methodical will break if moved. Install in the assets folder only and leave it there. Future versions will not have this limitation.

Version 1.1.1 Known issues

1. Throws a null reference error when the user gives an index of the wrong type when getting values from an array or list
2. Throws a null reference error when the user gives an index of the wrong type when getting values from a dictionary

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