24. April 2015: Unity Editor Extension Dynamic-Link Library

Table of Contents

1. What it is about	2
2. Why a dynamic-link library	2
3. What you need	2
4. Download the Unity project	3
5. Custom Editor with tooltips	4
6. Connect the Editor script with the standard script	5
7. Notice changes in the Inspector window	6
8. Add tooltips with description to intellisense	7
9. Hide public members from intellisense in the standard script	8
10. Use and Play the scene	9
11. Create a dynamic-link library for the Unity Engine & Editor	12
11.1. UnityEngine namespace	12
11.2. UnityEditor namespace	25
12. Test the .dlls / .xml in a new Unity project	29
13. Create a build application	30
14. Test the intellisense tooltips	
15. Automatically generate a script documentation	32

1. What it is about

In this workflow tutorial you will learn how to

- create a custom Editor with tooltips
- add a tooltip with description to intellisense
- hide public members from intellisense
- document the code
- create a dynamic-link library for the Unity Engine & Editor
- automatically create a documentation

↑ table of contents ↑

2. Why a dynamic-link library

To create a managed Dynamic-Link Library (.dll for short) has the following advantages

- public members, that are needed for the custom Editor, can be hidden from intellisense
- it's handier than dozens of scripts
- not everyone can easily read, edit and possibly steal the source code

↑ table of contents ↑

3. What you need

To complete this tutorial you'll need the following *free* tools: Unity with MonoDevelop to create the project, code, dynamic-link library and application DoxyGen to automatically create the documentation from code

Note: This tutorial was created with Unity 5.0.1 on a Windows 8.1 PC.

 \Uparrow table of contents \Uparrow

4. Download the Unity project

To download the project will help you to better follow and understand this tutorial.

Download abstractexample.unitypackage

Download the file, create a new project in Unity (menu bar -> File -> New Project...), call it e.g. "Unity Editor Extension .dll".

Import the Unity package in your new project (menu bar -> Assets -> Import Package -> Custom Package... -> *.unitypackage), then click the Import button in the window where the files are listed that are imported.

Double click / open the scripts *AbstractExample.cs* (in the *Scripts* folder of the project window) and *AbstractExampleEditor.cs* (in the *Editor* folder of the project window). Then open the scene *AbstractExampleScene* (in the main folder (Assets) of the project window).

In general you can also open MonoDevelop by using Unity's menu bar -> Assets -> Sync MonoDevelop Project.

Note: "Abstract" in this case has nothing to do with the C# keyword *abstract*. It only means that this is a theoretical example! I chose an abstract example to keep it as simple as possible.

5. Custom Editor with tooltips

A custom editor displays custom properties of your Unity standard script (called *AbstractExample.cs* in this example) in the Unity Inspector window.

To create a custom editor you need to extend your standard Unity script (that normally derives from the MonoBehaviour class) in MonoDevelop with a CustomEditor script (that derives from the Editor class and is called *AbstractExampleEditor.cs* in this example).

First you need to place this code inside the Editor script to tell the Editor script that it extends the standard script:

```
[CustomEditor( typeof( AbstractExample ))]
```

To show the public member variable

public Transform t_Abstract;

of the standard Unity script *AbstractExample.cs* in the Inspector window with a bold heading and tooltips you first need to connect this public member variable with the editor script. This is done by the following lines of code in the Editor script *AbstractExampleEditor.cs*:

```
private SerializedProperty sp_t_abstract;
```

```
private void OnEnable( )
{
    sp_t_abstract = serializedObject.FindProperty( "t_Abstract" );
}
```

If you now change the reference to $t_Abstract$ this will also automatically change the content of the variable $sp_t_abstract$.

Display the variable in Unity's inspector window by adding the following line of code to the editor script *AbstractExampleEditor.cs*:

EditorGUILayout.PropertyField(sp_t_abstract, new GUIContent("AbstractTransform", "Som..."));

The first string of *GUIContent* names the variable and the second string is the text shown by the tooltip when the name is hovered by the mouse cursor. The bold heading works the same way.

In the Inspector window hover the heading and the variable name to see the tooltips.

🔻 健 🗹 Abstra	act Example (Script)	💽 🌣,
Set the Tra	nsform here:	
AbstractTrAn	Drag and drop a gameobject from the Hierarchy	
	window into the rectangle below or click the small circle on the right side next to the rectangle and select a gameobject.	

6. Connect the Editor script with the standard script

In much cases the standard script needs feedback from the Editor script. For example if you need to change something when (like in this example) the reference (or value) of a variable was changed.

Note: When you create a build application of a Unity project the UnityEditor namespace and also the Editor script is cut away and is not included into the build application. So you need to place all code you possibly want to execute at the runtime of the build application (normally) inside Unity standard scripts.

To create a connection you need the following code inside the Editor script *AbstractExampleEditor.cs*:

private AbstractExample s_target;
private void OnEnable()
{
 s_target = (AbstractExample)target;
}

This creates a variable that references to the standard script AbstractExample.cs and with the line

```
s_target.AbstractTransformChanged( );
```

it's possible to call the public method *AbstractTransformChanged* of the class *AbstractExample* from within the class *AbstractExampleEditor*.

7. Notice changes in the Inspector window

To know when a change was made from the user, to call a method in the script, you need to create a "change check" that starts before the observed variable $sp_t_abstract$ in the code and ends behind it.

```
EditorGUI.BeginChangeCheck( );
EditorGUILayout.PropertyField( sp_t_abstract, new GUIContent( "AbstractTransform", "Som..." ));
if( EditorGUI.EndChangeCheck( ))
{
    serializedObject.ApplyModifiedProperties( );
    s_target.AbstractTransformChanged( );
}
```

If a change was made and *EditorGUI.BeginChangeCheck()* is true the change in *sp_t_abstract* is applied to *t_Abstract* with the first line of code within the *if* condition and then the method *AbstractTransformChanged* is called and several components are attached to the gameobject *t_Abstract* references to.

8. Add tooltips with description to intellisense

Add tooltips to the intellisense of public (or even private) members with this code:

```
/// <summary>Count the amount of components of a certain type that are attached to a gameo...
/// <para>T is the type of the component to count.</para>
/// </summary>
/// <param name="targetGameObject">The gameobject its certain type of compon...</param>
/// <returns>The amount of certain components attached to the gameobject.</returns>
/// <example>To call this method from anywhere use e.g.:
/// int numberOfBoxColliders;
/// numberOfBoxColliders = AbstractExample.GetComponentCount<BoxCollider&gt;( m... );
/// </code>
/// </example>
public static int GetComponentCount<T>( GameObject targetGameObject )
where T: Component
  Component[] componentTypeToCount;
  componentTypeToCount = ( Component[] )targetGameObject.GetComponents( typeof( T ));
  return componentTypeToCount.Length;
}
```

The /// comment slashes tell MonoDevelop (and also e.g. Microsoft Visual Studio) that these XML comments describe the member below in more detail.

<summary > should describe what the member is doing / good for.

<param name="parameterName"> describes a parameter of a method.

<*returns*> describes the content that is returned by the method.

<example>...<code> gives the user a multiline example how to use the member.

Note: Make sure you replace the pre-defined XML characters that the tooltips are displayed proper.

 \Uparrow table of contents \Uparrow

9. Hide public members from intellisense in the standard script

Some members in the standard script need to be public that the connected Editor script can access them. But in the final product the user should only see the methods that she/he is allowed to use and that are documented.

To hide these members just add this code above the member you want to hide from intellisense like this:

[System.ComponentModel.Browsable(false), System.ComponentModel.EditorBrowsable(System.ComponentModel.EditorBrowsableState.Never)] public Transform t_Abstract;

Note: To hide this variable from intellisense only shows that you can hide all kind of members. It is still accessible from outside.

Note2: This will only work in the build .dll! (So.. let's test it after the script was build to a .dll.)

10. Use and Play the scene

Let's test the example scene *AbstractExampleScene*. Follow the **screenshots** and their **descriptions** below how to do this.



Select the *AbstractController* gameobject in the Hierarchy window (1.). Now you can see the custom Editor in action. To connect the Cube gameobject to the script click the small circle (2.) in the Inspector window. A *Select Transfrom* window will show up. Here select the Cube gameobject (3.).

 Inspector 	<u> </u>	i Insp	pector				a	•≡
👕 🗸 Cube	🔲 Static 🛛 🛨	*	Cube			St	atic	•
Tag Untagged 🕴 Layer Default		Тад	Untagged	‡ La	yer Default			
▶ 👃 Transform	🖻 🗘	⊳ ,,	Transform	n			2	\$,
🕨 🧮 Cube (Mesh Filter)	🖻 🗘		Cube (Me	sh Filter)			2	\$,
🕨 🔜 🗹 Mesh Renderer	🛐 🌣,	▶ 🛃 🔽	Mesh Ren	derer			2	\$,
Default-Material	🕅 O.	🔻 📦 🔽	Вож Collid	ler			2	\$,
Shader Standard				🔥 Edit	Collider			
		Is Tri	igger					
Add Component		Mater	rial	None (Phy	sic Material)			
Add component		Cent	er	X 0	Υ 0	Z 0		
		Size		X 1		Ζ1		
		⊤ 🙏	Rigidbody				2	\$,
		Mass		1				
		Drag		0				
		Angu	lar Drag	0.05				
		Use (Gravity	\checkmark				
		Is Kir	nematic					
		Inter	polate	None				÷
		Collis	ion Detectio	on Discrete				
		Cons	traints •				-	-
			Constant	Force	V 0.01		131	₽,
		Polet	: ive Ferre	X 0.3	1 9.8L	20		
		Tora	ive Force	~ 0		20		
		Relat	ive Torque	XO	Y 0	7.0		
		Kelat	ave rorque					
			Default-N	1aterial			2	\$,
			Shader 🖪	Standard				Ţ

The method *AttachColliderRigidbodyAndHoveringConstForceIfNeeded* in the standard script *AbstractExample.cs* was called when the transform reference in the Inspector window was changed. (Left: before -- Right: after)



Click the Play button to run the scene. The Start method of the script *AbstractExample.cs* is called and generates the output in the console. The cube doesn't move.

If you would comment / change the lines 108 & 109 of code in the Start method like this

//DeleteColliderRigidbodyAndConstForceIfPresent(t_Abstract.gameObject); Debug.Log ("Components NOT deleted.");

the three attached components are not deleted and the Cube gameobject will move to the right on Play.

11. Create a dynamic-link library for the Unity Engine & Editor

To wrap the code in a .dll with MonoDevelop it is necessary to create two .dlls. One .dll includes the standart script and the other .dll includes the Editor script that will be cut off when the project is build as application. This' the reason why two .dlls are needed.

11.1. UnityEngine namespace

Create an new project in MonoDevelop and build the Editor script *AbstractExample.cs* as .dll. Follow the **pictures** and the **descriptions** how to do this.

<u>())</u>					Assembly-CS	harp ·	- Scripts\/	AbstractExample.cs -	MonoDevelop
File	Edit View Sea	arch Proje	ect Build	Run	Version Control	Tools	Window	Help	
	New	۲	📄 File		Ctrl+N	l o s	olution load	led.	
	Open	Ctrl+O	🗔 Solut	ion	Ctrl+Shift+N	<			
	Save	Ctrl+S	Work	space.		itor.cs	5 ×		
2	Save As		election						
Ø	Save All Ctrl	l+Shift+S			11-0-1 - 1-1-		н.		
5	Revert		actExample Singularity I	.cs" col Defensi	mpany="Singularity e All Rights Reser	/ Deten: ved <th>se"> opvright></th> <th></th> <th></th>	se"> opvright>		
	D		e						
	Recent Files		date>						
	Recent Solutions								
x	Close Solution								
×	Close	Ctrl+W	e finest abst	tract m	ethods inside. <th>nmarv></th> <th></th> <th></th> <th></th>	nmarv>			
٠	Quit	Ctrl+Q	iple : Mono	Behavi	our				
	14 { 13 #region public	r memher v	ariables						
	14	, member v	unubics						
	15 [System.C	Component	Model.Brow	vsable(false), System.Cor	nponen	tModel.Edit	orBrowsable(System.Con	nponentModel.Edit
	16 public Trans	storm t_Ab	stract;						
	18 #endregion pu	ublic memb	er variables						
	19								
	20 #region public 21	c static met	hods						
	22 /// <sumr< th=""><th>mary>Coun</th><th>t the amour</th><th>nt of co</th><th>mponents of a cer</th><th>tain typ</th><th>e that are at</th><th>tached to a gameobect.</th><th></th></sumr<>	mary>Coun	t the amour	nt of co	mponents of a cer	tain typ	e that are at	tached to a gameobect.	
	23 /// <para></para>	>T is the typ	pe of the co	mpone	nt to count. <th></th> <th></th> <th></th> <th></th>				
	24 /// 25 /// <th>imary></th> <th>arratCamaC</th> <th>biect"</th> <th>The appropriate it</th> <th>- cortai</th> <th></th> <th>moopoots are counted <!--</th--><th>(and the second s</th></th>	imary>	arratCamaC	biect"	The appropriate it	- cortai		moopoots are counted </th <th>(and the second s</th>	(and the second s

Create a new Solution.

/// <returns>The am /// <returns>The am /// <remarks>This m</remarks></returns></returns>	arget Gambooject 21ne gameooject is certain type or components are o jount of certain components attached to the gameobject. webod is very clever and i'm very proud of it so hav special attaction who New Soluti	ounteolsyparame
Recent Boo C# IL NUnit Other UnityScript VBNet	Console Project C# Empty Project C# Gtk# 2.0 Project C# Library E	Creates an empty C# class library
Name:	EditorExtensionStandardDLL	
Location:	E:\2014\T-8\GitHub	Browse
Solution name:	EditorExtensionStandardDLL	Create directory for solution
Project will be sa \EditorExtension targetGameObject] if(targetGameObject	aved at E:\2014\T-8\GitHub\EditorExtensionStandardDLL IStandardDLL t <i>gameObject.AddComponent<</i> BoxCollider>(]; t. <i>GetComponent</i> <rigidbody>(] == null)</rigidbody>	Cancel OK

In the *New Solution* window select that you want to create a $C# \rightarrow Library$, name it *EditorExtensionStandardDLL* and choose a proper location to save it. Then click the OK button.

<u>(</u>						EditorExt	ensio	nStandard	DLL -	- MyClass.cs	- MonoDevelop-U
File Ed	lit View	Search	Project	Build	Run	Version Control	Tool	Window	Help		
	Debug		~	Default		¥	0	MonoDevelo	p-Uni	ity	
	Debug		_		_						
- ▲ ▶	Releas	е	-	<	× (
No sele	ection	_									
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21	<pre>// // <auto-g <="" auto="" c="" chan="" co="" namespac="" o="" pre="" public="" runti="" syst="" the="" this="" using="" {="" }="" }<=""></auto-g></pre>	enerated ode was i me Versic ges to this ode is reg generated em; e EditorEx lass MyCl c MyClass	s file may enerated. l> densionS	d by a to 319.342 cause ir tandardi	ol. 09 ncorrec	t behavior and w	Il be lo	st if			

Change the build mode to *Release* in MonoDevelop because the source code is working in the Editor, so we assume it's also working as a build .dll.

orExte	nsionStand	ardDLL	- MyClass	s.cs - MonoD	evelop-	Unity		-	D	×	
ontrol	Tools Wind	low Hel	р								
~	O MonoD	evelop-Ui	nity				Q Press 'C	ontrol+,' to sea	rch		
and will	be lost if					Solu T	ution EditorExtension EditorExter EditorExter References System Assemblyir MyClass.cs	nStandardDLL IsionStandard Info.cs	DLL	□ ×	Properties 🔤 Doc
						Open With Version Con Open Conta Copy Cut Remove Rename Build Action Quick Prope Properties Refresh	htrol	<			cument Outline Solution

Now delete the automatically created file MyClass.cs in the solution window. (If you don't see the solution window tab you can let it appear with menu bar -> View -> Pads -> Solution.) We won't need it.

Edi	torExtensionStandardDLL - MonoDevelop-Unity			-	ð	×
ontrol	Tools Window Help					
~	Items saved.		Q Press 'Contr	rol+,' to search	٦	
		So ▼ □ ▼	Iution EditorExtensionSta EditorExtension References Assembly R	IndardDLL nStandardDL idit References Refresh		Properties Document Outline Solution

In the solution window click the small gear button and choose *Edit References...* to add references to other necessary .dlls.

R)	Edi	t References	×
All Packages Project	ts .Net Assembly	Q Search (Control+F)	Selected references:
🔰 👌 Program Fil	es Unity Editor Data	Managed <	System Version=4.0.0.0, Culture
Places <pre> Search Search Recently Used timow_001 Desktop Local Disk (C:) Local Disk (D:) Local Disk (E:) Removable Disk DVD RW Drive (G:) </pre>	Name A ICSharpCode.NRefa Mono.Cecil.dll Nunit.framework.dll Unity.CecilTools.dll Unity.DataContract.dll Unity.IvyParser.dll Unity.Locator.dll Unity.SerializationL UnityEditor.dll	Size Modified * 528,4 kB 10.04.2015 279,6 kB 14.04.2015 143,4 kB 10.04.2015 10,2 kB 14.04.2015 10,8 kB 14.04.2015 28,7 kB 14.04.2015 8,7 kB 14.04.2015 16,9 kB 14.04.2015 3,5 MB 14.04.2015	
BD-ROM Drive (H:)	UnityEditor.Graphs.dll	210,4 kB 14.04.2015	
÷ -	a onityEngine.di	Assemblies	
UnityEngine, Version=0	0.0.0.0, Culture=neutral	Add <-	Cancel OK

In the *Edit References* window change to the *.Net Assembly* tab and navigate to the *Managed* folder inside the Unity folder to select the *UnityEngine.dll*. Then add it to the *Selected references:* list on the right side. Click OK.

Project saved.	Press 'Control+,' to search
	Solution Solution
	 Build EditorExtensionStandardDLL Rebuild EditorExtensionStandardDLL Ctrl+F7 Clean EditorExtensionStandardDLL Run Item
New <u>F</u> ile	Start Debugging Item Set As Startup Project Add
Add Files Add Web Reference	Tools Version Control
<u>A</u> dd Files from Folder <u>A</u> dd Existing Folder	Find in Files Ctrl+Shift+F Open Containing Folder
	□ Copy Ctrl+C ✓ Cut Ctrl+X ☑ Delete

The next step is to link the source code to this solution. Click the small gear and click Add -> Add files...

<u>@</u>		Add file	es					×
	GitHub	Unity Editor Extensio	on .dll	Assets	Scripts	<	-	
Places Q Search Recently Used Timow_001	Name Abstr	actExample.cs				▲ Size 4,9 kB	Modified 15:12	^
 Desktop Local Disk (C:) Local Disk (D:) Local Disk (E:) Removable Disk 	Ξ							
 DVD RW Drive (G:) BD-ROM Drive (H:) Documents T-8 								
÷ •					All File	es		¥
Override default builder	ild action	None	Y		€8	ncet >	Open	

Browse to the Unity example project and select and open the file *AbstractExample.cs* in the Scripts folder.

 The file E:\2014\T-8\GitHub\Unity Editor Extension .dll\Assets \Scripts\AbstractExample.cs is outside the target directory. What would you like to do? Copy the file to the directory Move the file to the directory Add a link to the file <
 Copy the file to the directory Move the file to the directory Add a link to the file <
Move the file to the directory Add a link to the file < Cancel OK
Add a link to the file Cancel OK
Cancel OK

Select Add a link that the file is automatically updated here when you make changes in the Unity project.

		1	🖑 Properties 👘 Document Outline 🔄 Solution
×			• ×
- 0		arch	IDLL
-		Q Press 'Control+,' to sear	ution EditorExtensionStandardDLL Content ExtensionStandardI Content Extensi
			Solu V I E V I V
Develop-Unity	lelp		
ono	He	ed.	
Mo	Window	roject save	
	Tools	Ø P	
	ontrol	×	

The Solution window should look like this.

	Solution	× 🖂
	EditorExtensionStandardDLL	Pro
	> - EditorExtensionStandardDLL 💈	¢≁ Per
	Build EditorExtensionStandardDLL F7	ties
	Rebuild EditorExtensionStandardDLL Ctrl+F7	
	Clean EditorExtensionStandardDLL	
	Run Item	Docu
	Start Debugging Item	Ime
	Set As Startup Project	nt o
	Add	Utlii
	,	he
	Tools	
	Version Control	So
	Find in Files Ctrl+Shift+F	lutio
	Open Containing Folder	5
	Delete	
	Rename F2	
>	<u>O</u> ptions	
	Refresh	

To get tooltips it is necessary to change the compiler settings. Click the small gear and open the Options.

😡 Project Op	otions - EditorExtensionStandardDLL	Solution
▼ General		 EditorExtensionStan
Main Settings	Compiler	EditorExtension
▼ Build		References
🕌 General	Configuration: Release V Platform: Any CPU V	System
🕼 Custom Commands	Concert Onting	AbstractExample
Configurations	General Options	AssemblyInfo.cs
🔮 Compiler <====	Generate overflow checks	
🔅 Assembly Signing	 Enable optimizations 	
Output	Emit debugging information	
▼ Run	■> ✓ Generate xml documentation	
🖉 General	-	
🏟 Custom Commands	Define Symbols:	
Source Code	Platform target: Any CPU v	
INET Naming Policies		
Code Formatting	Warnings	
🕼 Standard Header	Warning Level: 4	
Name Conventions		
 Version Control 	ignore warnings:	
🕼 Commit Message Style	Treat warnings as errors	
ChangeLog Integration	Additional Options	
	Additional arguments:	
	Additional arguments.	
	Cancel OK	

In the *Project Options* window on the left side select *Build -> Compiler*. Make sure the *Configuration* is set to *Release*. Enable *Generate XML documentation* to generate the XML file with the documentation. MonoDevelop will use it to load the intellisense text. Then click the OK button.

<u>R</u>		EditorExtensionStandardDLL - MonoDevelop-Unity					
File Edit View Searet – Projer	Build Run Version	Control Tools	Window Help				
Release V	Build All Rebuild All Clean All		F8 Ctrl+F8				
>	🏭 Build EditorExter	nsionStandardDLL	F7				
	Rebuild EditorEx Clean EditorExte	tensionStandardDLL nsionStandardDLL	L Ctrl+F7				
	Stop		Shift+F5				

Now let's build the .dll. In the menu bar click *Build* -> *Build EditorExtensionStandardDLL*. Ignore the two warnings because the two members without XML comment will not show up in the intellisense at all. (Compare chapter 9.)

The build *EditorExtensionStandardDLL.dll* is saved into the MonoDevelop project folder under *bin* -> *Release* together with the *EditorExtensionStandardDLL.xml* file. Leave it there.. for now.

 \Uparrow table of contents \Uparrow

11.2. UnityEditor namespace

Create an new project in MonoDevelop and build the Editor script *AbstractExampleEditor.cs* as .dll. This works nearly exactly like to build the .dll of the standard script.

Choose *EditorExtensionEditorDLL* as name for this solution.

Follow the **screenshots** below, *that show only the differences to the standard script .dll creation process*, and the **descriptions** how to do it.

Q	Edit References ×
All Packages Project	ts .Net Assembly Q Search (Control+F) Selected references:
Places Places Search Recently Used Timow_001 Desktop Local Disk (C:) Local Disk (D:) Local Disk (E:) Removable Disk DVD RW Drive (G:) BD-ROM Drive (H:) T-8 2 Assemblies selected	es Unity Editor Data Managed System Version=4.0.0.0, Culture=neutr Name Size Modified ICSharpCode 528,4 kB 10.04.2015 Mono.Cecil.dll 279,6 kB 14.04.2015 Unity.CecilTool 10,2 kB 14.04.2015 Unity.DataCon 10,8 kB 14.04.2015 Unity.Locator.dll 8,7 kB 14.04.2015 Unity.Locator.dll 8,7 kB 14.04.2015 UnitySerializa 16,9 kB 14.04.2015 UnityEditor.Gr 210,4 kB 14.04.2015 UnityEditor.Gr 210,4 kB 14.04.2015 UnityEngine.dll 794,6 kB 14.04.2015
	Cancel OK

Also add a reference to the *UnityEditor.dll*.

R)		Edit Reference	s	×	
All Packages Project	IN I	Q Search (C bin Release	ontrol+F)	Selected references:	
 Recently Used timow_001 Desktop Local Disk (C:) Local Disk (D:) Local Disk (E:) Removable Di DVD RW Drive BD-ROM Drive T-8 T-8 EditorExtensionStandard 	 ConityEngine.dll rdDLL, Version=1.0.5 	Asse 5592.21673, Cultur	re=neAdd	C:\Program Files\Unity\Edi	

And add a reference to the .dll of the standard script that was build in chapter 11.1..

R	8							Add fil	les					×
	2	۸	2014	T-8	(GitHub	Unity Editor	r Extensi	on .dll	Assets	Editor	<	-	
	Places	5		~		Name				▲ Size		Modified		^
	🔍 Sea	arch	n			🗋 Abstra	actExampleEd	litor.cs	<	1,5	kВ	Yesterda	y at 20:48	
	🛞 Rec	ent	ly Used	d										
	៉ tim	ow_	001											
	🛅 Des	skto	р											
	🕹 Loc	al D	Disk (C:)										
	😞 Loc	al D	Disk (D:)										
	I Loc	al D	Disk (E:)	E									
	🥪 Rer	nov	able D	i										
	IV	D R	W Drive	ə										
	<i>.</i> ₿ BD∙	-RO	M Drive	ə										
	🛅 Doo	cum	ents											
	៉ T-8				IJ									
	🛅 Bac	kup	os	~										~
	-										All F	iles		*
	🗌 Ov	/erri	de defa	ault b	uile	d action	None		~					
											C	Cancel	Open	

Add a link to the file *AbstractExampleEditor.cs* instead of *AbstractExample.cs* to create a .dll of the file *AbstractExampleEditor.cs*.

EditorExtensionEditorDLL - MonoDevelop-Unity	- 🗇 🔁	k
ontrol Tools Window Help		
V Build successful.	Q Press 'Control+,' to search)
	Solution EditorExtensionEditorDLL EditorExtensionEditorDLL EditorExtensionEditorDLL UnityEditor.dll UnityEngine.dll EditorExtensionStandardDLL.dll System AbstractExampleEditor.cs AssemblyInfo.cs 	Properties Document Outline Solution

The solution window should look like this now.

You don't need to enable the *Generate XML documentation* option in the *Project Options* window because there was no documentation added to the file *AbstractExampleEditor.cs*. There shouldn't be any warning when you build the .dll.

12. Test the .dlls / .xml in a new Unity project

- 1. In Unity create a new project (*menu bar -> File -> New Project...*) and name it e.g. *EditorExtensionDLLTest*.
- 2. In the Project window create a folder called "Scripts" and another called "Editor".
- 3. Open the Windows File Explorer and navigate to the MonoDevelop project of the standard script (...*EditorExtensionStandardDLL\bin\Release*) and drag & drop the files *EditorExtensionStandardDLL.dll* and *EditorExtensionStandardDLL.xml* in Unity's Project window into the Scripts folder.
- 4. Use the Windows File Explorer to navigate to the MonoDevelop project of the Editor script (...*EditorExtensionEditorDLL\bin\Release*) and drag & drop the file *EditorExtensionEditorDLL.dll* in Unity's Project window into the Editor folder.
- 5. Add a new empty gameobject to the Hierarchy window with *menu bar -> GameObject -> Create Empty.*
- 6. In the Project window navigate to the Scripts folder and open the *EditorExtensionStandardDLL.dll* by clicking the small triangle to the left of the .dll. A file called *AbstractExample* will becomes visible.
- 7. Drag & drop the AbstractExample file to the empty GameObject in the Hierarchy window.
- 8. Add a cube to the Hierarchy window with *menu bar -> 3D Object -> Cube*.

Now you have the same scene setup like in the Unity example project with the source code files.

Note: Unity's new Plugin Inspector automatically detects the .dll correct. It's made for all platforms because it only contains "Unity code".

13. Create a build application

- 1. Attach the Cube gameobject to the GameObject gameobject to add the three components.
- 2. Check the Cube gameobject that all three components are attached by the .dll.
- 3. In the menu bar click File -> Build & Run.
- Set the Platform to "PC, Mac and Linux Standalone".
 (Btw. -- Aren't Mac OS X and Linux computers "personal computers" (PCs for short) as well ?
- 5. Click the "Build & Run" button and select a location to save the build and name it.
- 6. Click "Play!". (Press ALT + F4 to quit the application.)
- 7. The cube doesn't move so the .dll deletes the components when the application is started.

Note: When you have a look at the log file (<code>output_log.txt</code>) of the build application (located in the application folder) you'll notice that the number of attached BoxColliders seems to remain one! This happens because in the build application

GameObject.Destroy(targetsCollider);

is used, so the deletion is delayed slightly until after the current Update loop.

\Uparrow table of contents \Uparrow

14. Test the intellisense tooltips

To test the intellisense tooltips in MonoDevelop just use the Unity project from chapter 13. and create new C# script (or use the one included in the *abstractexample.unitypackage*) in the *Scripts* folder and type

AbstractExample.g

inside the method *Start* and you can see the description of the method *GetComponentCount*.

		Assembly-C	Sharp - Scripts	IntellisenseTest.	s* - MonoDeve	lop-Unity	
e Edit View Search	Project Build Run	Version Control	Tools Window	Help			
Debug	♥ Default	×	Solution loa	ded.			Q Press
IntellisenseT	est.cs 0	AbstractExampleE	ditor.cs ×	AbstractExample.cs	×		
IntellisenseTest 🕨	Start ()						
<pre>2 using System.Colle 3 4 public class Intelli 5 6 // Use this for in 7 vold Start () { 8 AbstractExam 9 } 10 11 // Update is cl 12 vold Update (13 14 } 15 } 16</pre>	ections; senseTest : MonoBehavio nitialization ple.g GetComponentCou	nt	Ge Ge J St Cd at cd	blic static int :tComponentCount <t> (;ameObject targetGam ummary punt the amount o tached to a game punt.</t>	eObject of components o obect. T is the t	of a certain typ ype of the corr	e that are ponent to

15. Automatically generate a script documentation

Open the *DoxyWizard* application.

Let's use the example project you have downloaded here. Then follow the screenshots and their **descriptions** below how to use *DoxyGen*.

8	Doxygen GUI frontend +	
File Settings Help		
Step 1: Specify the working directory from which doxygen will run	1	
C:/Users/timow_001/Desktop	Select	
Step 2: Configure doxygen using the Wizard and/or Expert tab, t	hen switch to the Run tab to generate the documentation	
Wizard Expert Run		
Topics	Provide some information about the project you are documenting	
Mode	Project name: Editor Extension .dll	
Output Diagrams	Project synopsis:	
	Project version or id:	
	Project logo: Select No Project logo selected.	
	Specify the directory to scan for source code	
	Source code directory: E:/2014/T-8/GitHub/Unity Editor Extension .dll Select	
	✓ Scan recursively <	
	Specify the directory where doxygen should put the generated documentation	
	> Destination directory: C:/Users/timow_001/Desktop Select	
	Previous Next	

In the *Wizard* tab fill out the *Project* Topic and make sure you enable recursive scanning.

8	Doxygen GUI frontend + -	
File Settings Help		
Step 1: Specify the working directory from which doxygen will run		
C:/Users/timow_001/Desktop	Select	
Step 2: Configure doxygen using the Wizard and/or Expert tab, th	nen switch to the Run tab to generate the documentation	
Wizard Expert Run		
Topics		
Project	Select the desired extraction mode:	
Mode	 Documented entities only 	
Output	All Entities	
Diagrams	Include cross-referenced source code in the output	
	Select programming language to optimize the results for Optimize for C++ output Optimize for C++/CLI output Optimize for Java or C# output Optimize for C or PHP output Optimize for Fortran output Optimize for VHDL output	
	Previous Next	
[]		

In the Mode *Topic* select *C*# *output*.

e engliser eer nemene	
File Settings Help	
Step 1: Specify the working directory from which doxygen will run	
C:/Users/timow_001/Desktop	Select
Step 2: Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documentation	
Wizard Expert Run	
Topics	
Project Select the output format(s) to generate	
Mode Mode	
Output	
Diagrams O with navigation panel	
prepare for compressed HTML (.chm)	
Vith search function	
Change color	
as intermediate format for hundrinled 005	
as intermediate format for hyperinked PDP	
as intermediate format for PDF	
 as intermediate format for PostScript 	
Man pages	
Rich Text Format (RTF)	
L XML	
Previous	> Next

In the *Output* Topic disable *LaTeX*. Don't change anything in the *Diagrams* Topic.

File Settings Help		
Step 1: Specify the working directory from which doxygen	will run	
C:/Users/timow_001/Desktop		Select
Step 2: Configurydoxygen using the Wizard and/or Expert	t tab, then switch to the Run tab to generate the documentation	
Wizard Expert Run	n da z Bonalan olan horne hala eta eta eta da da zen zen dala bionen barraten 200 eta horne bizten.	
Breiert	EXTRACT_ALL	^
Build	EXTRACT_PRIVATE	
Messages	EXTRACT_PACKAGE	
Input	EXTRACT_STATIC	
Source Browser	EXTRACT_LOCAL_CLASSES	
Index	EXTRACT_LOCAL_METHODS	
HTML	EXTRACT_ANON_NSPACES	
LaTeX	HIDE_UNDOC_MEMBERS	
RIF	HIDE_UNDOC_CLASSES	
Vian V	HIDE_FRIEND_COMPOUNDS	
HIDE_UNDOC_MEMBERS	HIDE_IN_BODY_DOCS	
If the WEDD WINDOW WINDOW IS SHOULD BE	INTERNAL_DOCS	
doxygen will hide all undocumented members inside	CASE_SENSE_NAMES	
documented dasses or files. If set to NO these members	HIDE_SCOPE_NAMES	
will be included in the various overviews, but no documentation section is generated. This option has no	HIDE_COMPOUND_REFERENCE	
effect if EXTRACT_ALL is enabled.	SHOW_INCLUDE_FILES	
The default unles in 10	SHOW_GROUPED_MEMB_INC	
The default value is: NO.	FORCE_LOCAL_INCLUDES	
	INLINE INFO	~
	Previous	> Next

Switch to the *Expert* tab and select the *Build* Topic. Enable the *HIDE_UNDOC_MEMBERS* and *HIDE_UNDOC_CLASSES* flags that all classes and members without XML comment are not included into the documentation.

3	Doxygen GUI frontend +		- 🗆 🗙
File Settings Help			
Step 1: Specify the working directory from which doxygen will run	1		
C:/Users/timow_001/Desktop			Select
Step 2: Configure doxygev sing the Wizard and/or Expert tab, t	hen switch to the Run tab to generate the documentation		
Wizard Expert Run			
Run doxygen Status: not running		Show configuration	Save log
Output produced by doxygen			
<pre>Cenerating namespace member index Generating annotated compound index Generating alphabetical compound index Generating hierarchical class index Generating member index Generating file index Generating file member index Generating example index finalizing index lists writing tag file lookup cache used 3/65536 hits=8 misses=4 finished E:/2014/T-8/GitHub/Unity Editor Extension AbstractExampleEditor is not documented. E:/2014/T-8/GitHub/Unity Editor Extension AbstractTransformChanged() (function) of cl E:/2014/T-8/GitHub/Unity Editor Extension t_Abstract (variable) of class AbstractExam E:/2014/T-8/GitHub/Unity Editor Extension t_Abstract (variable) of class AbstractExam E:/2014/T-8/GitHub/Unity Editor Extension OnInspectorGUI() (function) of class AbstractExam *** Doxygen has finished</pre>	.dll/Assets/Editor/AbstractExampleEditor.cs .dll/Assets/Scripts/AbstractExample.cs:47: lass AbstractExample is not documented. .dll/Assets/Scripts/AbstractExample.cs:16: mple is not documented. .dll/Assets/Editor/AbstractExampleEditor.cs actExampleEditor is not documented.	:13: warning: Co warning: Member warning: Member :20: warning: Me	mpound mber
Show HTML output			

Change to the *Run* tab and click the *Run doxygen* button. When *DoxyGen* has finished click the *Show HTML output* button.

Editor Extension .dll Intervention SubtractExample Class Reference Only be treat adotact methods more. Hore Public Member Functions Vertion Extension addition of the constant force component for the generobyed. Vertion Extension Class Reference Public Member Functions Vertion Extension Class Reference Vertion Extension Class Reference Public Member Functions Vertion Extension Class Reference Vertion Extension Class Reference Vertion Extension Class Reference Public Member Functions Vertion Extension Class Reference Static Public Member Functions Extension Extensio	← → C III II file///C/Users/timov_001/Desktop/html/class_abstract_example.html		
Main Page Casses Casses Casses AbstractExample Class Reference Casses Only the inext dostact inethods inside. Mare Interferent dostact inethods inside. Mare Public Member Functions Casses vist AtschooliderRigidbodyAndHoveringConstForceRMeeded (CameCbyect targetCameCbyect) Casses Vist Atschoolider RigidbodyAndHoveringConstForceRMeeded (CameCbyect targetCameCbyect) Casses Vist Atschoolider RigidbodyAndHoveringConstForceRMeeded (CameCbyect targetCameCbyect) Casses Vist Atschoolider RigidbodyAndHoveringConstForceRMeeded (CameCbyect targetCameCbyect) Casses Atschoolider RigidbodyAndHoveringConstForceRMeeded (CameCbyect targetCameCbyect) Casses	Editor Extension .dll		
Cate Control Cate Control Control Contro Control	Main Page Classes		
Catholic Catholic Catholic AbstractExample Class Reference			
AbstractExample Class Reference Only the treest stotact methods ussde. More thethet MonoReleavour. Public Member Functions vest AtschedingerRightbody/and/loweringConstForcetReeded (ClampDoject targetClampDiject). Attaches a box colleer, the rightbody and the constant force component to this gameobject. More vest Attaches a box colleer, the rightbody and the constant force components that are attached to this gameobject. Parameters targetCampDiject The gameobject a box colledr, the rightbody and the constant force components that are attached to this gameobject. More Static Public Member Functions static rightDomponenties of a certain type that are attached to a gameobject. More Output the amount of components of a certain type that are attached to a gameobject. More Detailed Description Owner Function Documentation vest AbstractExample AttacheOliderRightbodyAndHoveringConstForcetReeded (SameObject targetGameObject) Attaches a box collider, the rightbody and the constation tops gameobject. Vest AbstractExample AttacheOliderRightbodyAndHoveringConstForcetReeded (SameObject targetGameObject) Vest AbstractExample AttacheOliderRightbodyAndHoveringConstForcetReeded (SameObject targetGameObject) Vest AbstractExample AttacheOliderRightbodyAndHoveringConstForcetReeded (SameObject targetGameObject) Ve	LENCE LATE LENCE PROPERTY.		
Output the treat databation methods inside. Mate Interts MonoBehaviour. Public Member Functions vist AttachColliderRigitg/bodyAndHoveringConstForceINeeded (EameObject TargetCameCtglect) Attaches a box coller, the rigitbody and the constant force component to this gameobject. More vist Detection ColliderRigitg/bodyAndHoveringConstForceINeeded (EameObject TargetCameCtglect) Attaches a box coller, the rigitbody and the constant force component to this gameobject. More vist Detection ColliderRigitg/bodyAndHoveringConstForceINeeded (EameObject Three component Is deleted from. More Static rel Count the amount of components of a certain type that are attached to a gameobect. More Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Count the amount of componentSouries T> (ChameObject TargetCameCtglect) Attaches a box collider, the rigidbodyAndHoveringConstForceINeeded (CameObject targetGameCtbject) Attaches a box collider, the rigidbodyAndHoveringConstForceINeeded (CameObject targetGameObject) Attaches a box collider, the rigidbodyAndHoveringConstForceINeeded (CameObject targetGameObject)	AbstractExample Class Reference		
Interests MonoBehaviour: Public Member Functions vist AttachColliderRigidbodyAndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a box collider, the rigidbody and the constant force components that are attached to his gameobject. More vist DeleteColliderRigidbodyAndConstForcellReesent (GameCbyect targetGameCbject) DeleteColliderRigidbodyAndConstForcellReesent (GameCbyect More Static Public Member Functions static GetComponentCourst T > (GameCbyect targetGameCbject) Court the amount of components of a certain type that are attached to a gamecbect. More vist AttachColliderRigidbodyAndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a bas cellider, the rigidbody AndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a bas cellider, the rigidbody AndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a bas cellider, the rigidbodyAndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a bas cellider, the rigidbodyAndHoveringConstForcellReeded (GameCbyect targetGameCbject) Attaches a bas cellider, the rigidbody and the censtant force component to this gameobject DeleteColliderRigidbodyAndHoveringConstForcellReeded (GameCbyect targetGameCbject)	Only the linest abstract methods inside. More		
Public Member Functions visit AttachColliderRigidbodyAndHoveringConstForcelRiseaded (CameObject targetCameObject) Attaches a box collete, the rigidbody and the constant force components to this gameobject. More visit DeterectiveRigidbodyAndHoconstForcelRiseant (CameObject targetCameObject) DeterectiveRigidbodyAndHoconstForcelRiseant (CameObject TargetCameObject) DeterectiveRiseant box collete, the rigidbody and the constant force components that are attached to this gameobject. Parameters targetCameObject The gameobject targetCameObject targetCameObject) DeterectiveRiseant for constraint force components in a re attached to this gameobject. Static Public Member Functions static Int GetComponentCounst T > (CameObject targetCameObject) Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description only the freed abstract methods inside word AbstractExample.AttachColliderRigidbodyAndHoveringConstForcelRiseded (GameObject targetGameObject) Attaches a box colleter, the rigidbody and the constant force component to this gameobject Object targetGameObject targetGameObject targetGameObject targetGameObject (GameObject targetGameObject)	Inherits MonoBehaviour.		
vels AttachColliderRigidbodyAndHowingConstForcetINeeded (GameObject targetCameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject. More vols DetectColliderRigidbodyAndConstForcetINeeded (GameObject targetCameObject) Detetes one box collider, the rigidbody and the constant force component is deteted from. More Static Public Member Functions static int GesComponentCounts T > (CameObject targetCameObject, More Detailed Description Only the frest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoverngConstForcetINeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and HoverngConstForcetINeeded (GameObject targetGameObject)	Public Member Functions		
void DeteteColliderRigidbodyAndConstPorceIfPresent (GameObject targetGameObject) Detetes one box collider, the rigidbody and the constant force components that are attached to this gameobject. Parameters targetGameObject The gameobject a box collider, the rigidbody and the constant force component is deleted from. More Static Public Member Functions static infl_GetComponentCounts TP (GameObject targetGameObject). Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the frest abdract methods inside Member Function Documentation Void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceIfNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component of this gameobject. More Void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceIfNeeded (GameObject targetGameObject) ImagetGameObject) Attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	void AttachColliderRigidbodyAndHoveringConstForceitNeeded (GameObject largetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject. More		
Parameters targetGameObject The gameobject a box collider, the rigitbody and the constant force component is deleted from. More Static Public Member Functions static Int_GetComponentCount< T > (CameObject targetCameObject) Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigitbodyAndHoveringConstForceItNeeded (GameObject_targetGameObject) Attaches a box collider, the rigitbody and the constant force component to this gameobject only attaches a component if the gameobject doesn't have a component to this type attached yet. Parameters	void DeleteColliderRigidbodyAndConstForcelfPresent (GameObject targetGameObject) Deletes one box collider, the rigidbody and the constant force components that are attached to this gameobject.		
Image: Construction Image: Construction of a certain type that are attached to a gameobect. More Static Public Member Functions Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the finest abstract methods inside: Member Function Documentation Image: Component Counter Rigidbody AndHoveringConstForceitNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component of this gameobject Count the amount of the rigidbody and the constant force component of this gameobject. Member Function Documentation Image: Component of the rigidbody and the constant force component to this gameobject. Image: Component of the rigidbody and the constant force component to this gameobject. Count the amount of component of the rigidbody and the constant force component to this gameobject. Image: Component of the rigidbody and the constant force component to this gameobject. Count targetGameObject targetGameObject (Counter Counter Counte	Personatans		
More Static Public Member Functions static Intl GetComponentCounts T > (CameObject TargetClameCbject) Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceitNeeded (GameObject TargetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only attaches a component if the gameobject doesn't have a component to this gameobject Parameters	r memory is a second to be a second		
More Static Public Member Functions static IntGetComponentCounts T > (GameDaject targetClameDbject) Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the treast abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceltNeeded (GameObject targetGameObject) Attaches a bax cellider, the rigidbody and the censiant tone component to this gameobject Only attaches a component if the gameobject doesn't have a component to this type attached yet. Parameters	Angeleaning appearing a provide a box contrar, we have an an excitating the component is denoted from.		
Static Public Member Functions static Int GetComponentCounts T > (GameObject TargetGameObject) Count the amount of components of a certain type that are attached to a gameobject. More Detailed Description Only the freest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForce#Needed (GameObject_targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only attaches a component it the gameobject doesn't have a component to this gameobject Parameters	More		
static int GetComponentCount T > (GameObject largetQameObject) Count the amount of components of a certain type that are attached to a gameobect. More Detailed Description Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForce#Needed (GameObject targetGameObject) Attaches a box collder, the rigidbody and the constant force component to this gameobject. Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	Static Public Member Functions		
Detailed Description Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceitNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	static int GetComponentCount< T > (GameObject targetGameObject) Count the amount of components of a certain type that are attached to a gameobect. More		
Detailed Description Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForce#Needed (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters			
Only the finest abstract methods inside Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceltNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	Detailed Description		
Member Function Documentation void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceliNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject Only allaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	Only the finest abstract methods inside.		
void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceltNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject. Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	Member Function Documentation		
void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceltNeeded (GameObject targetGameObject) Attaches a box collider, the rigidbody and the constant force component to this gameobject. Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters			
Attaches a box collider, the rigidbody and the constant force component to this gameobject. Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	void AbstractExample.AttachColliderRigidbodyAndHoveringConstForceltNeeded (GameObject targetGameObject)		
Only attaches a component if the gameobject doesn't have a component of this type attached yet. Parameters	Attaches a box collider, the rigidbody and the constant force component to this gameobject.		
Parameters	Only attaches a component if the gameobject doesn't have a component of this type attached yet.		

The result.

Congratulations! You have finished this tutorial successfully. :)