

# Intro to Unity

Hands-On Challenges

# Introduction to Unity Hands-On Challenges

Copyright © 2014 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.

# Challenge G: Getting your scripting sealegs

If you choose to build a game in Unity, you will be doing a whole bunch of scripting. While you can certainly do a whole lot of customization with interface along with plugins, there will be a time when you will need a tweak things to behave in a certain manner.

Thankfully, Unity comes with the three different scripting options and your first option is to pick the language that you will be using.

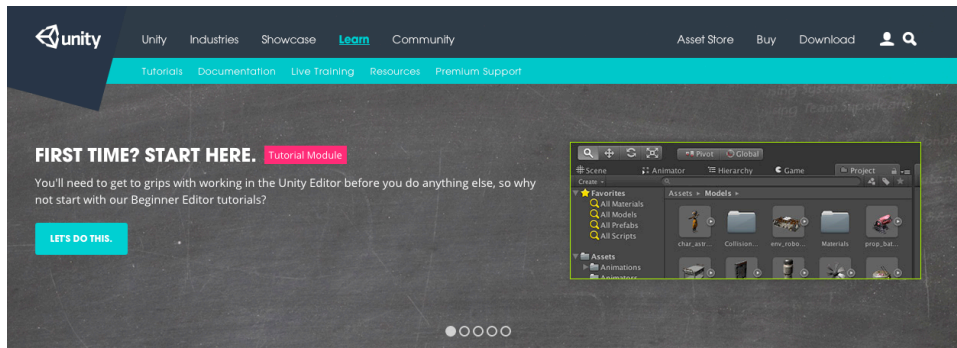
## Getting started

Where's the best place to learn about these languages? The documentation, of course!

Start by heading over to this URL:

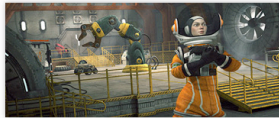
<http://unity3d.com/learn>

At the time that this document was written, the documentation looks like the following:



## LEARN WITH UNITY

There are many ways to learn Unity. In these pages you'll find everything you need to become a Unity developer. So why not start learning and join the community today? In [Tutorials](#) you'll find video and article based content, our [Documentation](#) are a complete written manual and scripting reference, and if you'd like some time with our experts, sign up for a [live Q&A session](#) and ask your questions directly.



### TUTORIALS →

Video and text tutorials in an array of topics &



### DOCUMENTATION →

Explore Unity's Component reference and browse

## RECENT TUTORIALS

### [Game Over](#)

Beginner - Survival shooter

### [More Enemies](#)

Beginner - Survival shooter

### [Scoring points](#)

Beginner - Survival shooter

### [Harming Enemies](#)

Beginner - Survival shooter

### [Player Health](#)

Beginner - Survival shooter

### [Health HUD](#)

Beginner - Survival shooter

To learn about the differences between C# and Javascript, a good place to look is their free video tutorials. Click on the **Tutorials block**, and from there, select **Scripting**.

Under the **Beginner** category, you should see a video on **C# vs JS syntax**. There are plenty of other videos to watch as well.

Unity also releases plenty of live training videos. Each week, Unity releases a new video that has a wealth of information such as creating a 2D platformer or writing your own plugins.

There are two videos called [Scripting Primer and Q&A](#) and [Scripting Primer and Q&A - Continued](#). Give these a watch as they will illuminate some aspects of how scripting works in Unity.

Unfortunately, there's not a lot of information on Boo so you will have to do some digging in Google to find some up-to-date resources.

## Reading the documentation

Once you have watched some of the videos and have a feel of how scripting does work inside of Unity, take a look over at the API documentation for GameObjects.

Head back to the Unity learning hub over here:

<http://unity3d.com/learn>

Select the **Documentation** module. You should be presented with two options: the **Unity Manual** and the **Unity Scripting API**.

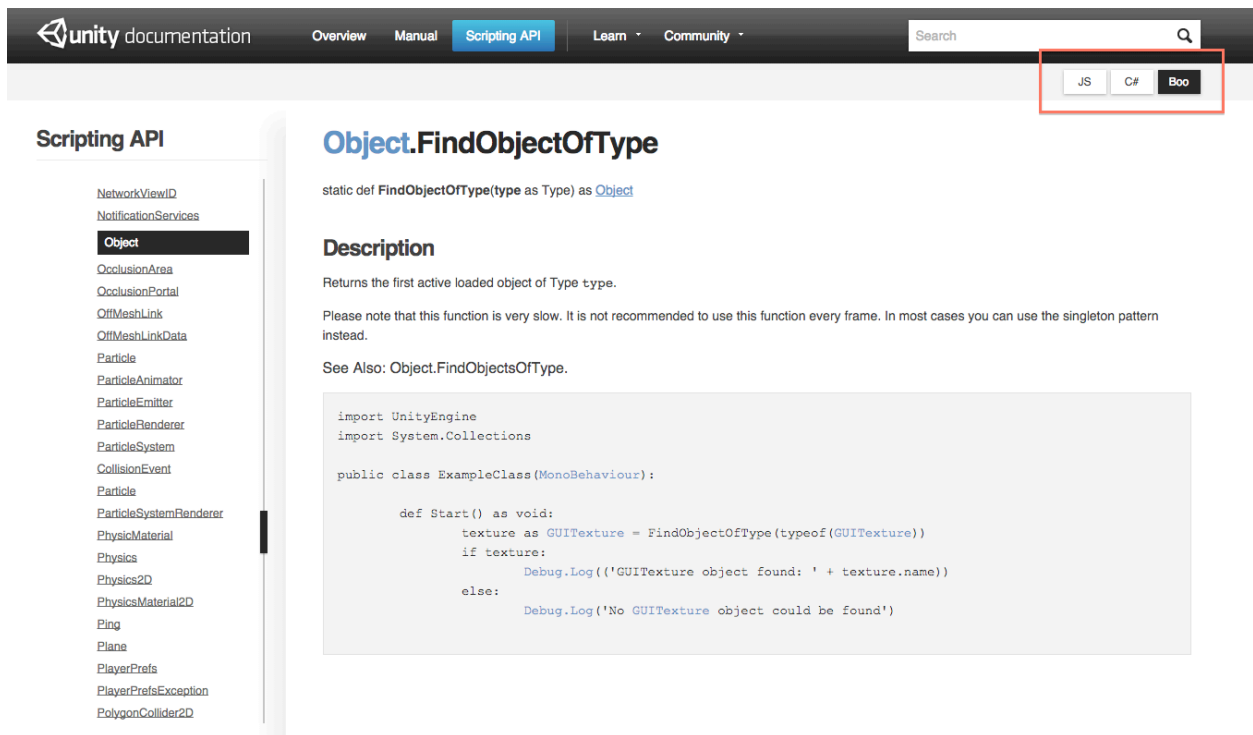
The manual provides a high level overview of the various components inside of Unity whereas the Unity Scripting API lists all the methods and properties as well provides plenty of code examples.

Select the **Unity Manual**, and open the **Unity Overview** category. Next, open **Building Scenes**, and then select **GameObjects**. There you will find lots of information regarding GameObjects and how they can be used inside of the engine.

Return back to the documentation hub, and select the **Unity Scripting API**. From the **Scripting API** list, select **UnityEngine**, then open the **Classes**. From the classes, find the **GameObject** class and take a look at the API. You will see that you have lots options when working with it.

Since Unity provides three different languages, you may find the Unity documentation using code samples in a different language. Thankfully, you do have a way to change this.

Just underneath the navigation bar, you will see a segmented list providing you the three different options. Select the language that you will be using, and the documentation will update accordingly.



The screenshot shows the Unity documentation website. At the top, there is a navigation bar with the Unity logo and the text "unity documentation". To the right of the logo are links for "Overview", "Manual", "Scripting API" (which is highlighted in blue), "Learn", and "Community". A search bar is located on the right side of the navigation bar. Below the navigation bar, there is a segmented list for language selection with three buttons: "JS", "C#", and "Boo". The "Boo" button is currently selected and highlighted with a red box. The main content area is titled "Scripting API" and "Object.FindObjectOfType". It includes a description of the method, a code example in C#, and a list of related classes in the left sidebar.

unity documentation Overview Manual **Scripting API** Learn Community Search

JS C# **Boo**

### Scripting API

- NetworkViewID
- NotificationServices
- Object**
- OcclusionArea
- OcclusionPortal
- OffMeshLink
- OffMeshLinkData
- Particle
- ParticleAnimator
- ParticleEmitter
- ParticleRenderer
- ParticleSystem
- CollisionEvent
- Particle
- ParticleSystemRenderer
- PhysicMaterial
- Physics
- Physics2D
- PhysicMaterial2D
- Ping
- Plane
- PlayerPrefs
- PlayerPrefsException
- PolygonCollider2D

## Object.FindObjectOfType

static def FindObjectOfType(type as Type) as Object

### Description

Returns the first active loaded object of Type type.

Please note that this function is very slow. It is not recommended to use this function every frame. In most cases you can use the singleton pattern instead.

See Also: Object.FindObjectsOfType.

```
import UnityEngine
import System.Collections

public class ExampleClass(MonoBehaviour):

    def Start() as void:
        texture as GUITexture = FindObjectOfType(typeof(GUITexture))
        if texture:
            Debug.Log('GUITexture object found: ' + texture.name)
        else:
            Debug.Log('No GUITexture object could be found!')
```

## Adding a Script

Now that you have an idea of how Scripting should work, it's time to add a script to your project. In the video tutorial, I demo'd the creation of a light changing script. Your challenge is to add that script to your breakout game.

Open the starter challenge project. If you've been following this series from the beginning, open your last saved project.

**Note:** Unity will open to an empty scene. Find the scene that you saved in your project view, or if you are using the Starter Project, open Main.scene.

In your scripts folder, create a new script called **RandomLightColor**. Finally, watch the video to see the script and enter it into your code editor. In a later video, you will be adding it to your lights to see this in action.

Welcome to the world of scripting! I hope you made yourself comfortable because you'll be spending a lot of time here! :]