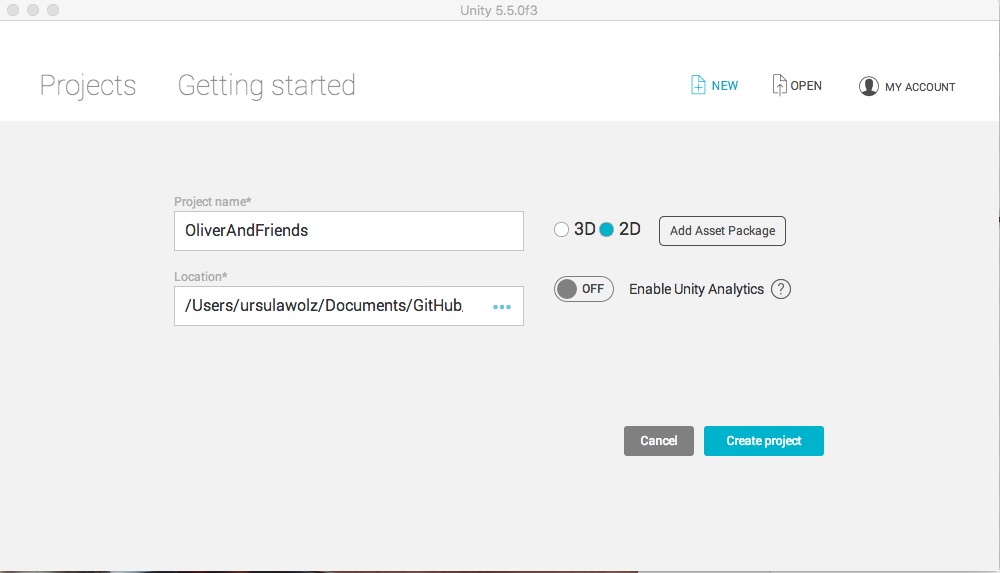
Genderless in Germany

**Introduction to Unity Objects**

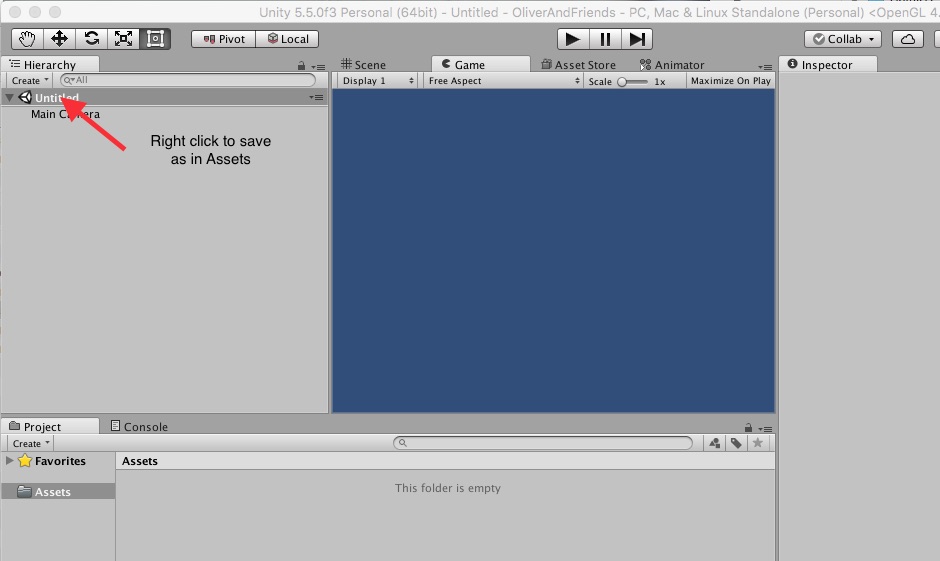
December 2016

**Overview:** This short tutorial introduces you to the basic building blocks of Unity: the *component.*  This tutorial also introduces the fundamental ideas of *object-oriented programming.*  This is a broad rather than deep introduction, showing you only what you need to get started[[1]](#footnote-1)A component is just an OBJECT. It has data and methods associated with it. You build components through the Unity IDE. Note that you can also build components through code. Follow these instructions:

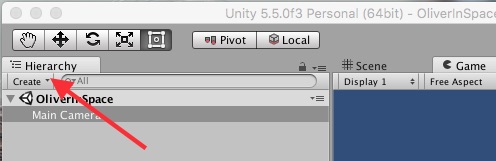
1. Create a 2D project in your personal space called OliverAndFriends. (See tutorial 0).

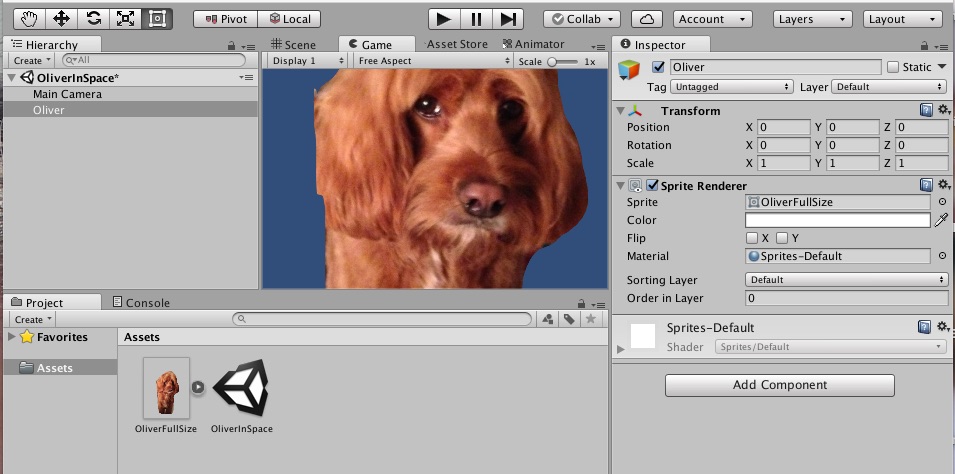
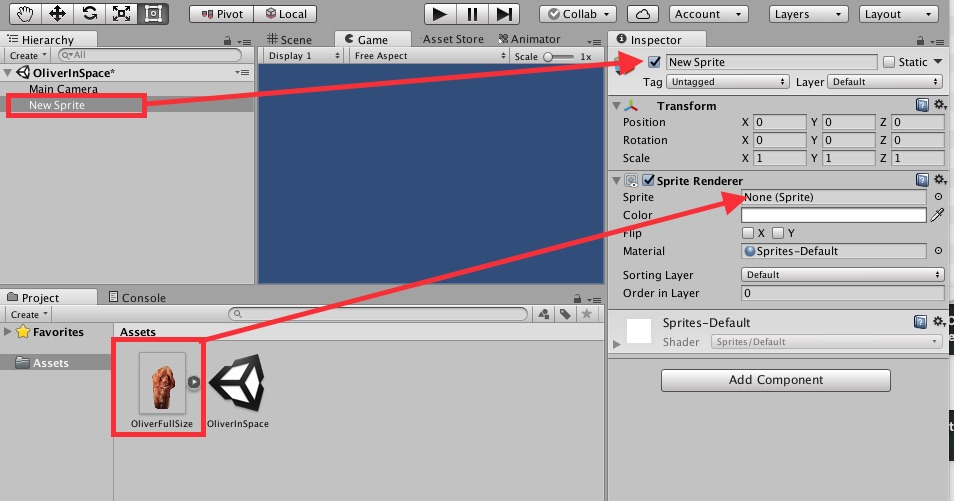


1. Your Unity project has an untitled scene and a main camera in that scene. Locate the Hierarchy window, right click the Untitled scene, and **save it as** ‘OliverInSpace’ in your assets folder. Get in the habit of putting components in scenes from the start. This hierarchy is a ‘has-a’ hierarchy. A project *has* a collection of scenes, each of which has components. Add another scene, save it, and notice your assets now contains two scenes. Go to your operating system and look at the folder that contains your project. Study the hierarchy of information contained in you’re the folder. In Unity, remove the second scene from both the hierarchy and the assets[[2]](#footnote-2).



1. Click on the OliverInSpace icon in the Asset window. Notice that the inspector shows you the details of the scene – there aren’t any yet. But the scene does have one component: the main camera. Click on the **main camera** component in the hierarchy and see what its details are in the inspector. See if you can figure out how to change the background color. Notice that the projection is Orthographic. This is necessary for 2D. You can learn more about this from the Unity tutorials later.
2. Add a 2D Object (sprite) to the scene via the *create* button in the hierarchy window.



1. The Misc Assets folder contains a file OliverFullSize.png. This is a Photoshop picture that is huge on purpose. You can simply drag it into the Assets window (on a Mac). Notice that this creates a new sprite in the hierarchy. The Inspector should show you that your sprite has two components of its own: a Transform and a Sprite Renderer. If not, click on the New Sprite component in the Hierarchy. The Sprite renderer is not connected yet. Drag the Oliver asset to the slot in the Sprite Renderer, and change the sprite name directly in the inspector. Oliver will appear in the Game window. He is huge. You can scale him down in the Transform. Experiment with his position, scale and rotation. Note that you want to rotate around the z axis.
2. 
3. In your OS, look at the size of OliverFullSize. Compare it with the Oliver.png asset. The latter was reduced outside of Unity to create an asset that is significantly smaller. This can be critical to how your project loads and how it updates. The full size Oliver was used here primarily to show you what not to do.
4. Your turn: Add some other sprites to game. You can use mages from the Misc. Assets folder or create your own. Notice that once an asset is added to a project, a copy is made in the project folder.
5. Finally, take a look at whether you are viewing the Game window or the Scene window. Switch to the Scene window in which you can move your sprites. Above the Hierarchy tab are buttons to control the Transform component. Experiment with theses a bit. Note that your 2D is actually positioned on the Z axis, so you can get some interesting effects. For those using this tutorial beyond GenderlessInGermany, you will want to explore 3D down the road. You can reset the transform via its gearbox in the upper left corner of the inspector.

HD:Users:ursulawolz:Desktop:GrabImages:Unity01:Buttons.jpg

1. .

**Object Oriented Wrap Up:**  Unity uses objects to represent all of the assets needed to create a project. We will often refer to the project as a game, although Unity is really an interaction engine rather than merely a game engine. Scenes are objects that have components such as cameras and 2D Objects. But the ‘create’ button suggests that there are other kinds of objects that can be added to a scene. Depending on how you played around in this project you have created *instances* (sometimes called *instantiations)* of classes of objects such as cameras, scenes and 2D Objects. You have specific ones that are based on *class definitions* that are part of Unity. All modern programming that involves graphical user interfaces (GUIs – pronounced Gooyes) is dependent upon classes of objects. Without them, we would be stuck in a world of text.

1. In our experience the Unity tutorials dive deep, showing you far too much at one time. This tutorial focuses on the essential principles, giving you a chance to build things up slowly. Of course you are welcome to go off an explore on your own at any point. [↑](#footnote-ref-1)
2. There is also a way to duplicate a scene. See if you can figure it out, and create a one page tutorial for it. [↑](#footnote-ref-2)