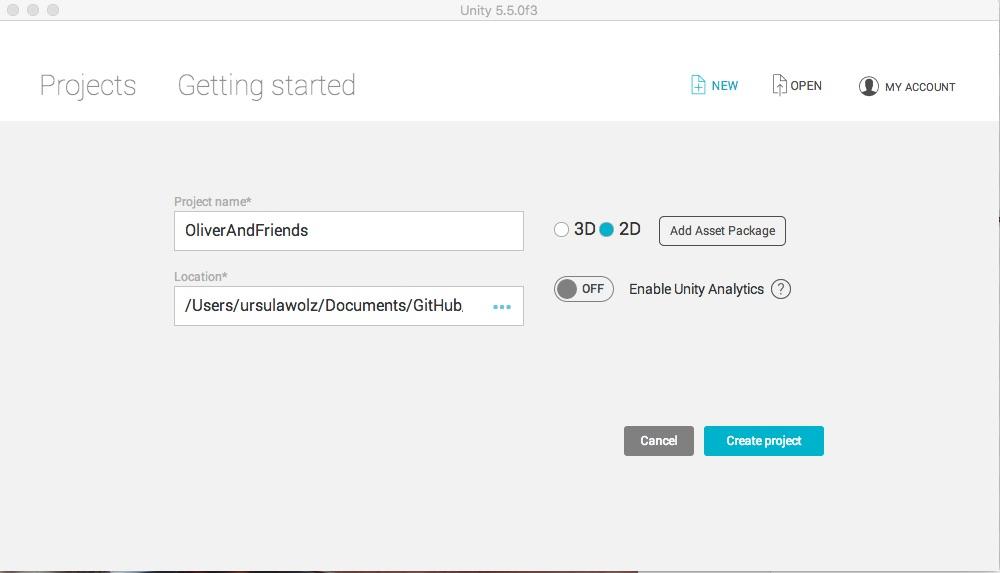
**Topic: Objects: Scenes, Game Objects, Components**

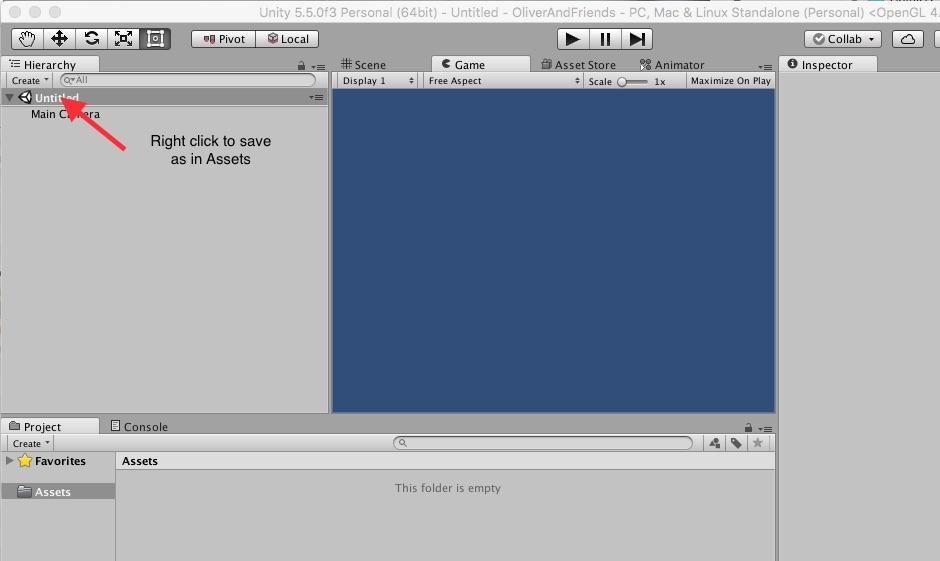
**Overview:** This is an introduction to basic building blocks of Unity3D: t*he scene, game object* and *component.*  It also demonstrates the fundamental idea of *object-oriented programming (OOP),* that a software system is made up of a set of *modules* called *objects.* When code is executed, *instances* of those objects are created, manipulated, referenced for information and (in well-written code) destroyed when no longer needed. Think of an object as a blueprint and an instance as an actual implementation of that blueprint. OOP provides organizational mechanisms to help the programmer manage complexity of both how parts of the code interact, and how efficiently the code can be executed. OOP is a great example of *modular design*: complex code is broken up into self-contained, small objects.

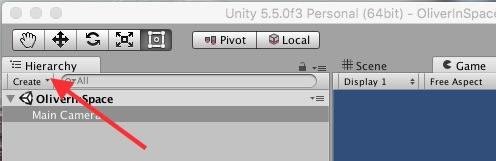
This tutorial also shows you how to create scenes, objects, and their components. More thorough treatment of these ideas can be found in the Unity3D [Overview](https://docs.unity3d.com/Manual/UnityOverview.html) and [2D overview](https://docs.unity3d.com/Manual/Unity2D.html).

**Your Turn (Please do the following):** It is assumed you have installed a version of Unity on your computer and understand how the development environment is organized. (See ‘Navigating the Environment’)

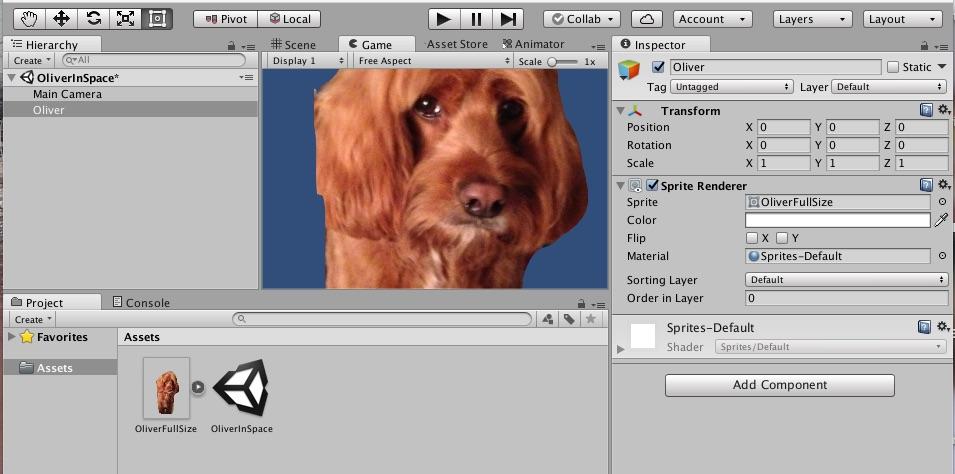
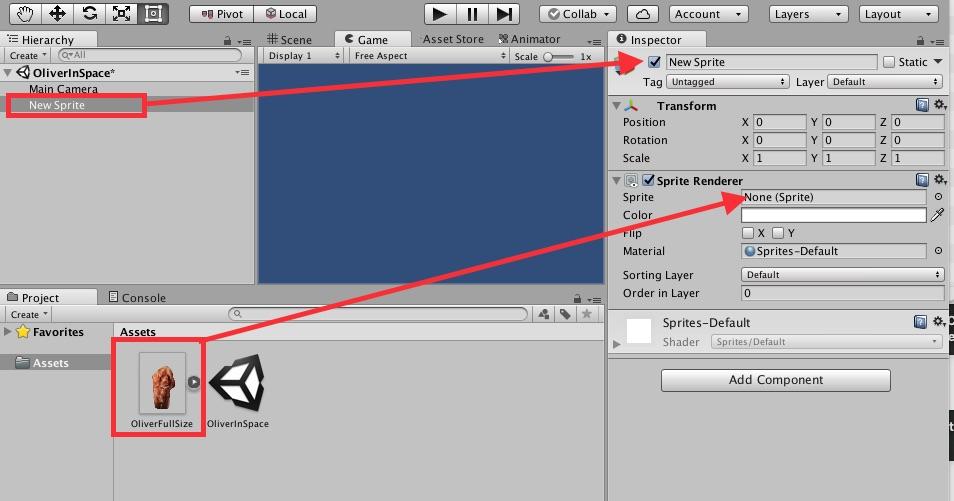
1. Create a 2D project in your personal space called OliverAndFriends



1. Your Unity project has an untitled scene and a main camera in that scene. Locate the Hierarchy panel, right click the Untitled scene, and **save it as** ‘OliverInSpace’ in your assets folder. This hierarchy is a ‘has-a’ hierarchy. A project *has* a collection of scenes, each of which has game objects, each of which has components. Add another scene, save it, and notice your assets folder now contains two scenes. Go to your operating system and look at the folder that contains your project. In the Unity3D environment, remove the second scene from both the hierarchy and the assets[[1]](#footnote-0).   
     
     
   
2. Click on the OliverInSpace icon in the Asset panel. 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 its components in the inspector. Figure out how to change the background color. Notice that the projection is Orthographic. This is helpful, but not necessary for 2D. See the the Unity3D reference materials for more information.
3. 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 rotate around the z axis because we are in 2D.



1. In your OS, look at the size of OliverFullSize. Compare it with the Oliver.png asset. The latter was reduced outside of Unity (in Mac Preview) 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 to illustrate how the Transform component (which is a separate object from the Oliver sprite can be directly manipulated in the inspector).
2. Add some other sprites to scene, using images from Misc Assets or your own. . Notice that once an asset is added to a project, a copy is made in the project folder so that all project information is contained in a folder.
3. Finally, take a look at whether you are viewing the Game window or the Scene window. Switch to the Scene panel in which you can change transform attributes of the sprite. Above the Hierarchy tab are buttons to control the Transform component. Experiment with theses a bit. You can reset the transform via its gearbox in the upper left corner of the inspector.HD:Users:ursulawolz:Desktop:GrabImages:Unity01:Buttons.jpg

**Object Oriented Wrap Up:**  The SampleProjects folder contains a completed version of OliverAndFriends. Unity3D uses objects to represent everything needed to create a project. Scenes are objects that have components such as cameras and 2D Game Objects. The ‘create’ button in the Hierarchy panel suggests that there are other kinds of objects that can be added to a scene. You were asked to create *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 Unity3D. All modern programming that involves graphical user interfaces (GUIs – pronounced Gooyes) is dependent upon classes of objects. Without them, the complexity of managing a software project would be overwhelming for programmers.

1. There is also a way to duplicate a scene. See if you can figure it out, [↑](#footnote-ref-0)