**Mastering Unity Project Folder Structure. Level 2 – Assets Organization**

[11.03.2014](http://developers.nravo.com/mastering-unity-project-folder-structure-level-2-assets-organization/) Posted by [Vitaliy Zasadnyy](http://developers.nravo.com/author/vitaliy-zasadnyy/)

Без рубрики



Have you ever faced a problem that it was hard to find assets in your project? If you’re new to Unity and make game more complex than [Space Invaders](https://www.google.com/search?q=Space+Invaders&oq=Space+Invaders), the answer is probably yes. That’s exactly what happened with our team, we are working on a big 3D MMO game and three months ago we decided that our project requires restructuring. In this post I’ll show folder structure that we use, explain how it is connected with art production pipeline and describe some naming conventions that help us a lot.

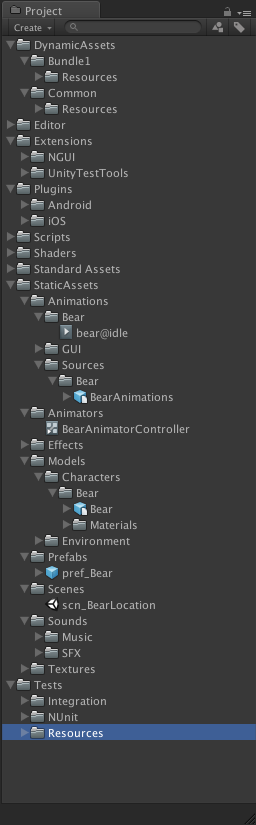
After a few hours of googling I found several topics about project structure best practices:

* <http://www.glenstevens.ca/unity3d-best-practices/>
* <http://team3dadiu.wikispaces.com/Unity+Project+Structure>
* <http://devmag.org.za/2012/07/12/50-tips-for-working-with-unity-best-practices/>

All these posts describe very simple cases (suitable for small games) and some basic rules of assets organization. So we decided to create our own set of rules and now, after several iterations of folder restructuring we are ready to share them.

**Assets Folder Structure**

When we started restructuring we had two simple goals: root folder should be as clean as possible and prepare folder structure to use [Asset Bundles](https://docs.unity3d.com/Documentation/Manual/AssetBundlesIntro.html). Below you can find a screenshot from our project window and explanations to some folders.



/DynamicAssets/[asset bundle id]/Resources – place to put assets dynamically loaded via Resources.Load(...) method. Common folder contains only base resources packed with game, resources from other folders are compiled into Asset Bundles and loaded from our servers on demand. Note, if you’re not using asset bundles you can only have one Resources folder.

/Extensions/[extension name] – by default, all third party extensions like NGUI are unpacked to the Assets folder root, but after fifth extension you’ll get mess in the project, so we have decided to create a separate folder.

/StaticAssets – when we’ve just started our game we used folder structure described [here](http://devmag.org.za/2012/07/12/50-tips-for-working-with-unity-best-practices/), after four months of development we realized that it’s total mess in the root, so we moved all static resources (referenced directly from scenes) to separate folder.

/StaticAssets/Animations – in order to better understand hierarchy related to animations I’d like to describe a part of our art pipeline (note, we use [Mecanim](https://docs.unity3d.com/Documentation/Manual/MecanimAnimationSystem.html)):

* animator imports animations into Unity in .fbx format and puts them under /StaticAssets/Animations/Sources/[model name] folder
* animator duplicates all the .anim files from .fbx, configures them and moves to /StaticAssets/Animations/[model name]
* animator creates mecanim animator controller for specific model, puts it under /StaticAssets/Animators, configures mechanim state machine using .anim files from /StaticAssets/Animations/[model name] folder.

/StaticAssets/Effects – folder for models, materials, textures, prefabs, etc. used for particle systems.

**WARNING**

**Don’t use spaces in file and folder names.** We faced with this problem during project build setup on CI server, as we found Unity3D [command line tools](https://docs.unity3d.com/Documentation/Manual/CommandLineArguments.html) can’t automatically process paths with spaces.

**Naming conventions**

* Append prefixes to differentiate asset types, e.g. pref\_ for prefabs, scn\_ – for scenes. We started to use prefixes when we discovered that it is easy to confuse prefab with model. Another advantage of this approach is instead of typing t:Prefab Bear in search field you type pref\_Bear, which is shorter
* Folder structure under /DynamicAssets/[asset bundle id]/Resources is exactly the same and almost the same as under /StaticAssets

More naming conventions you can find in this [post](http://www.glenstevens.ca/unity3d-best-practices/) under *“Naming Standards and Folder Structure”* section.

More posts from Mastering Unity Project Folder Structure series:

* [Level 0 – Folders required for version control systems](http://developers.nravo.com/mastering-unity-project-folder-structure-level-0-vcs)
* [Level 1 – Reserved Folders](http://developers.nravo.com/mastering-unity-project-folder-structure-level-1-reserved-folders)