**UW Animation Capstone, VR**

Unity 2018.1 Post-Processing and Shading Graph

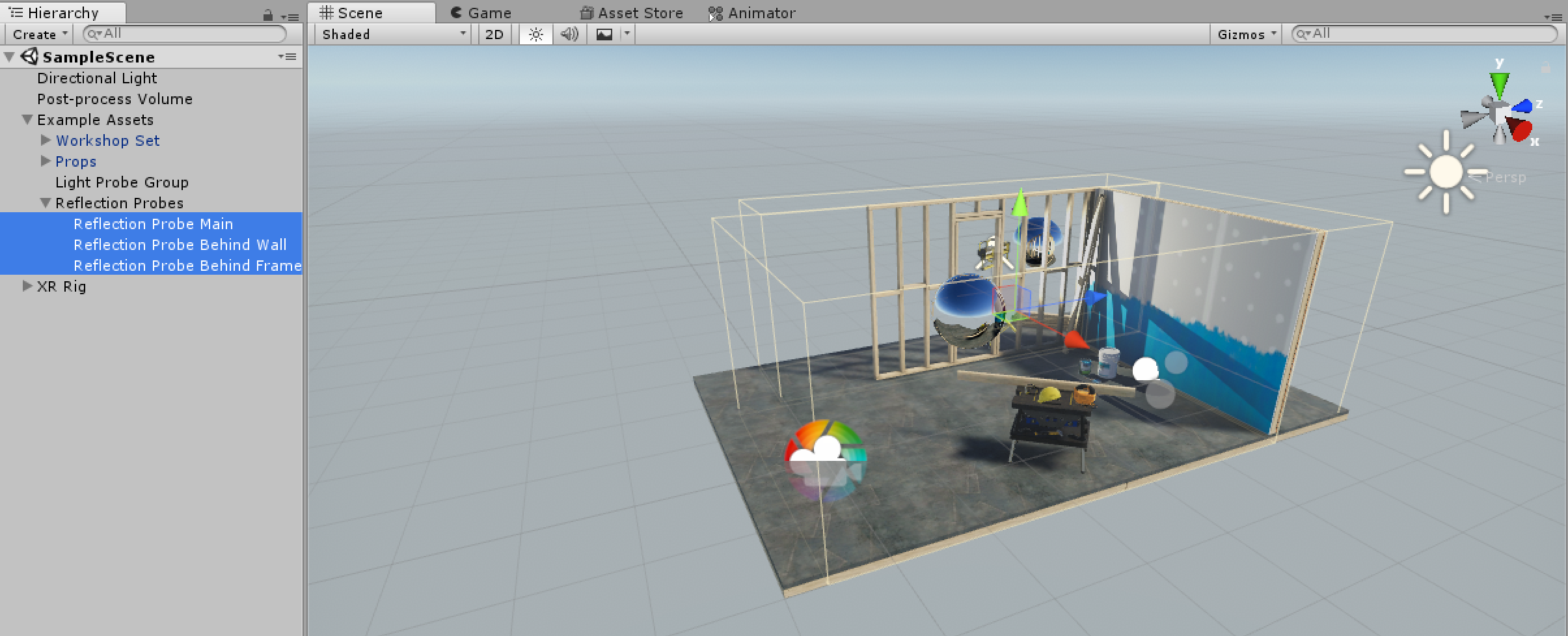
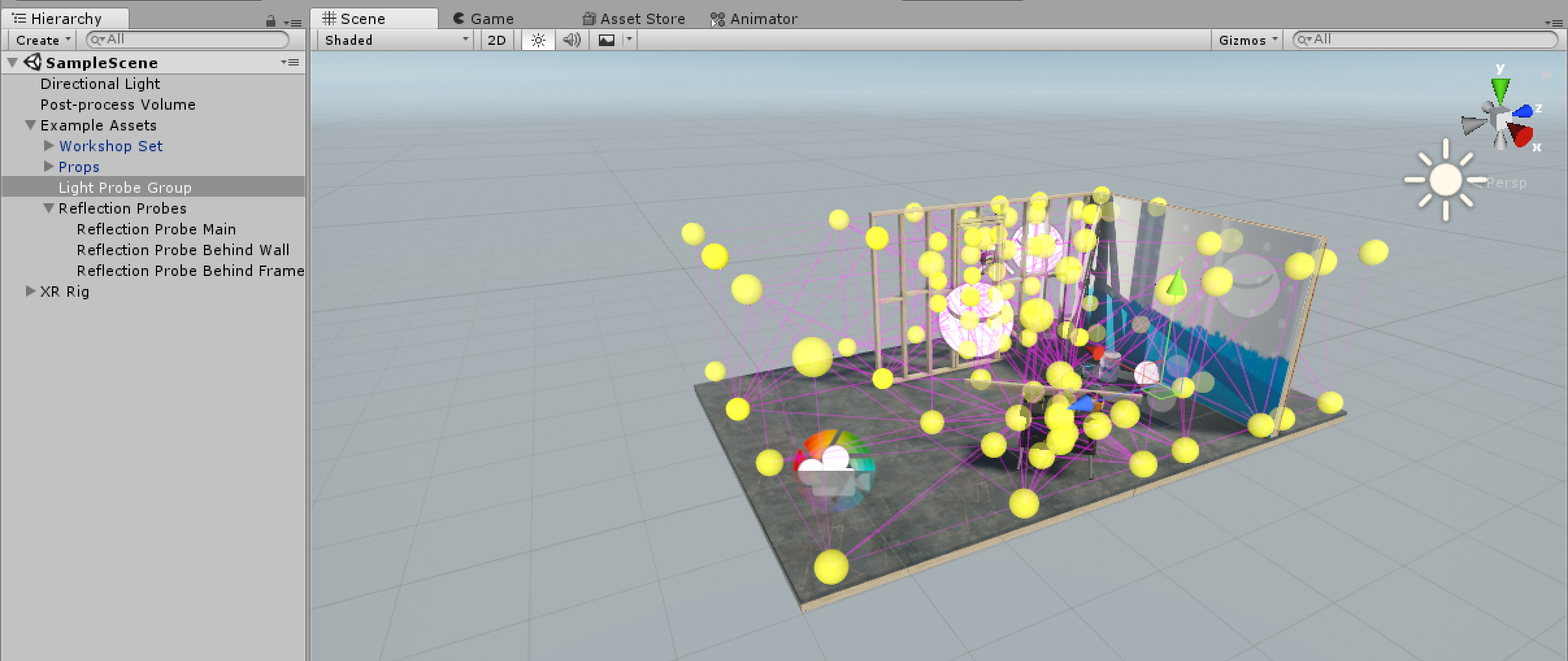
Demoed by Natalie Burke, March 29, 2018

**Disclaimer:** Links ARE NOT up to date with 2018.1. Please only use as general reference. Images are from 2018.1b13

1. Set up the project (choose one of the below)

To create a new VR project starting in Unity 2018.

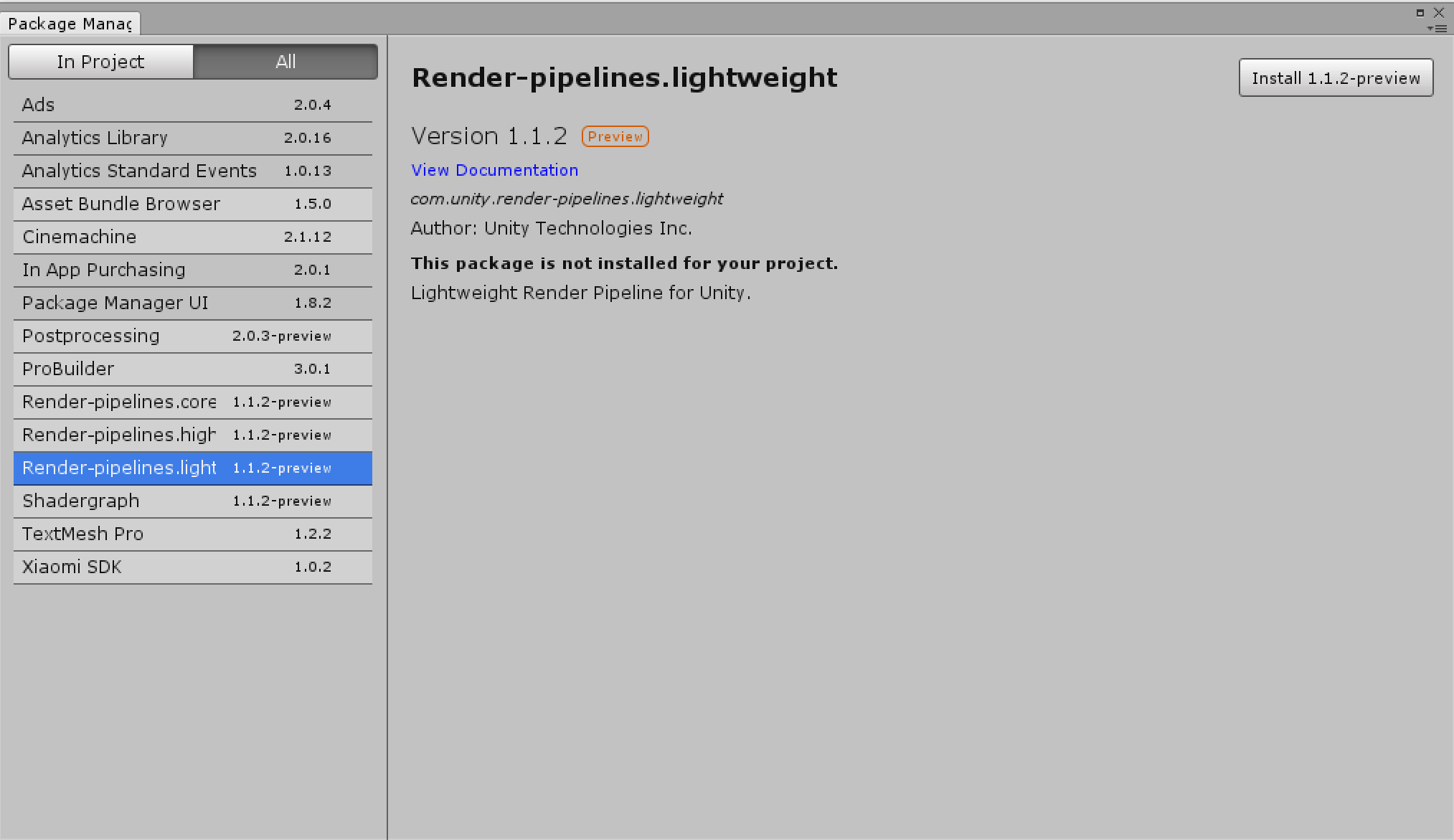
* 1. Create new project. Several new templates were added in 2018. Use “Lightweight RP for VR”



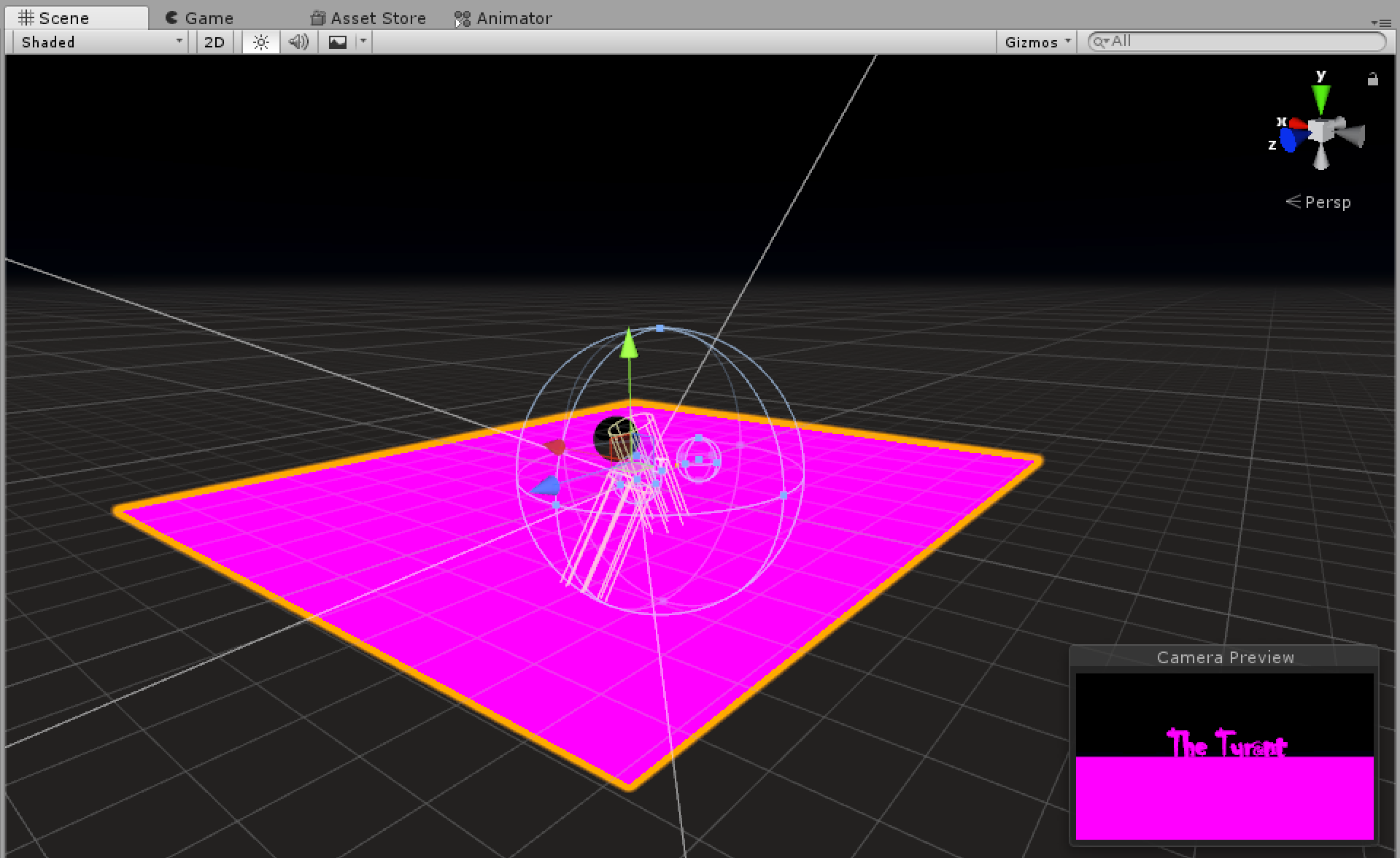
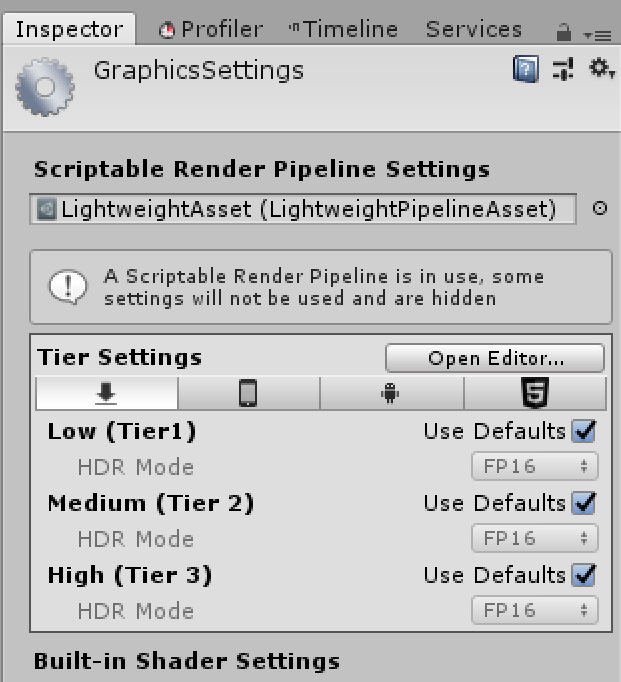
(Scenes come with example content, including a **light probe** setup (left, generates indirect light data based on probes) and **post-process volumes** (right, has presets with texture import and audio settings, if using the VR template will include settings tuned for VR)

To update to 2018.1 from old Unity version

* 1. Install lightweight render pipeline
     1. Open package manager: **window > package manager**
     2. Render-pipelines.lightweight install



* 1. Change rendering to lightweight
     1. Open settings folder (create one if it doesn’t exist) and right click: **create > rendering > lightweight pipeline asset.** This should create a LightweightAsset
     2. **Edit > project settings > graphics.** In the inspector, make sure the **Scriptable Render Pipeline Settings** is set to the new LightweightAsset. Everything should turn pink



* + 1. Select all materials in the scene (click and drag over the scene). **Edit > render pipeline > upgrade selected materials to lightweight.** Click proceed.
  1. Change light **render modes**. Only one light (the main shadow light, usually a directional one) should be real-time, all others should be baked.
     1. To change render mode, click on the light. In the inspector, change the mode to baked

1. Setting up initial post-processing.

Overview: <https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessingOverview.html>

* 1. Create post-processing volume (artist-placeable zones with automated transitions)
     1. Create asset: **create > 3d object > post-processing volume**
     2. Camera needs to be physically inside the post-process volume to work (Ctrl-shift-f to move camera)
  2. Set **post-processing layers** for volumes (<https://docs.unity3d.com/2018.1/Documentation/Manual/Layers.html>)
     1. Click on **post-processing volume**. In the inspector, click on the “layers” dropdown and click “create new”
     2. Add “post-processing” layer
     3. Go back to the inspector for the post-processing volume. Make sure the “layers” dropdown is set to “post-processing.” Repeat for all such volumes
  3. Set post-processing layers for cameras (<https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-Stack-SetUp.html>)
     1. Click on the camera object. Click on “add component” and add a post-processing component.
     2. Make sure post-processing component has the same tag as the volumes (“post-processing”)
  4. Set **overrides** for post-processing volumes (recommended ones for VR below)
     + Color grading (altering the color luminescence - <https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-ColorGrading.html>). Make sure mode is set to ASCC.
     + Bloom (simulates image artifacts, e.g. a bright light - <https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-Bloom.html>)
     + Auto-exposure (to account for eye adjustment for light setup)
     + Vignette (darkening of the edges - <https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-Vignette.html>)
  5. Overrides to AVOID for VR
     + Depth-of-field (<https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-DepthOfField.html>) – render times, doesn’t work too well for VR
     + Grain (<https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-Grain.html>) - want to be HD as possible for VR
     + Motion blur (https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-MotionBlur.html ) - finicky for VR
     + Screen-space reflections (<https://docs.unity3d.com/2018.1/Documentation/Manual/PostProcessing-ScreenSpaceReflection.html>) - because of render times

1. Setting up shading graph
   1. Create PBR graph
      1. Create new shading graph (in assets, right click, create > shading > PBR graph)
      2. What it should look like initially:
   2. Create nodes
      * Right click > create node > create node
      * Checkerboard node example below:
2. General notes for nodes
   * Textures: Can click and drag from the folders into shader graph
   * Color of wires indicate the type, change in color indicates type change, e.g. a 4-vector to 3-vector
   * Normal nodes: the type will be “normal,” rather than default
   * Preview node can be used to preview attributes
3. Attaching shading nodes to an asset
   1. Create material
   2. Click and drag node editor (?)
4. Using the Blackboard (Can create properties and animate values to them)
   1. Click plus sign to create property and set name
   2. Set values
5. Propagating texture to materials
   1. Select all nodes, right click, and convert to subgraph. This creates one node for everything (double click to enter a copy of the original graph)
   2. Attach as property in the inspector