Guidelines for art

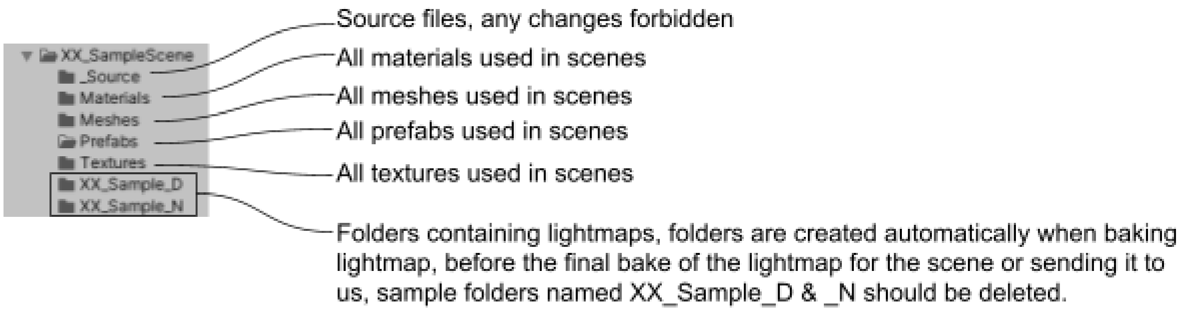
creating assets, naming, general instructions

NOTE: you need to request an identifier from us for your project. This identifier identifies your stage in the environment. Please fill the form on our website and we will get back to you with an identifier.

When you receive the identifier, ask your Unity developer to replace the “XX” in this spec with that number.

NOTE: these instructions accompany a Unity package (a ZIP file) that you need to provide to your Unity Developer. This package includes the test scenes for day and night. It also includes some things you may not need. These items were created for sound camps and they include laser lights, fire, and a sound-reactive logic that you can use to sync the tune of the music to certain things in the environment.

1. **Requirements for Unity:**
   - Your project will be required to meet all of the standards in Unity version at least 2019.3.10 and Universal Rendering Pipeline (URP)
   - All assets will be delivered as a one unity package, which will keep all of its features after unpacked
   - Unity package will include level with one main prefab containing all other meshes with materials and textures (nested prefabs if needed)
   - To keep importing packages as clean as possible, packages should be nested under a folder with proper installation name as the title.
   - Assets should be properly sorted into folders with English titles. Packages should not have a range of file types included under a singular folder
   - All files used in the project should be stored in one folder (Exhibit Scenes/XX_Xxxxxx). Folder hierarchy should look like XX_SampleScene, “XX” is the unique identification number given by our company.



Source files, any changes forbidden

All materials used in scenes

All meshes used in scenes

All prefabs used in scenes

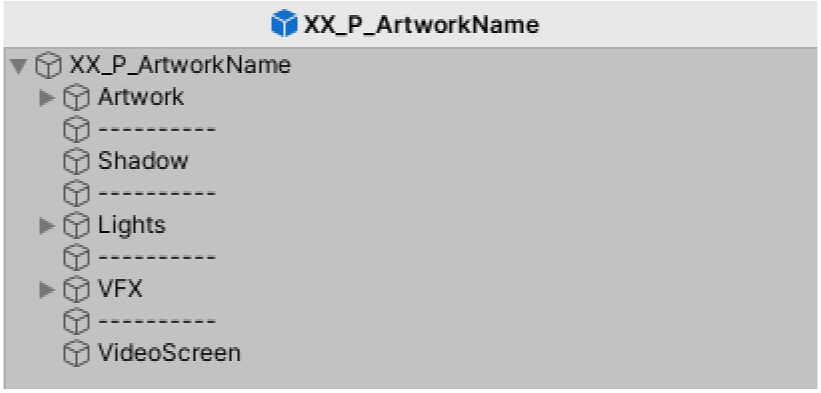
All textures used in scenes

Folders containing lightmaps, folders are created automatically when baking lightmap, before the final bake of the lightmap for the scene or sending it to us, sample folders named XX_Sample_D & _N should be deleted.

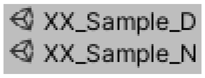
- There should be not any third party plugin being used(excluding Amplify Shaders and others that don't have any impact on game engine or creates files that do not require its installation for proper functionality)
- After downloading our example *.unitypackage create blank URP project, then using Package Manager (Menu Bar/Window/Package Manager) install Visual Effects Graph. This one is used to create some VFX like FlameThrower and gives you more freedom when creating VFX in scenes. (Default Unity Particle System can also be used)
  Then import our example package.

2. **Art Content:**

- **Scene** files are stored in main folder - XX_SampleScene
  Because the application will use the time of day, you have to create two scenes, day and night versions.
- **Scene naming.** The artwork scene should have names:
- Scene construction have to be based on **Main Prefab -** contains all objects used to create artwork, named XX_P_ArtworkName, stored in Prefabs folder. (every prefab should have a name - **XX_P_xxxxx**)



- Under **Artwork** GameObject - place all objects that are used for the basic structure of the art installation.

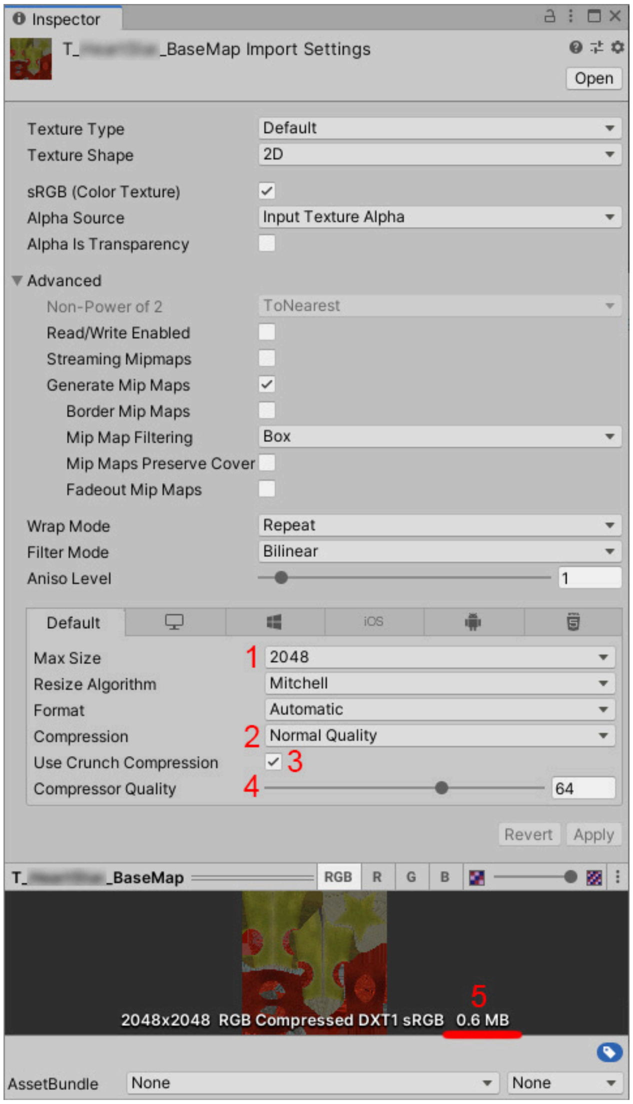


- **Shadow** GameObject is just a plane (is used to store the shadow information), it is a necessary object for the correct appearance of the scene, in your project it should have a default material, the shadows baked on it and the glow may look unattractive, but after the final file is delivered, the textures will be adjusted by us to extract only shadows and glow from the lightmap. When creating the scene, the scale of the "Shadow" object should be adjusted to the scale of the exhibit. If your stage has a lot of small objects spaced apart, it would be a good practice to create many "Shadow" objects to save space on the lightmap.
- Meshes you can name just as **XX_meshName**; **XX_AMP_name -** for Amplify Shaders;
- Place all lights under **Lights** GameObject. (Other objects with emissive material can be placed under Artwork)
- Use **VFX** GameObject for all Particle systems and other VFXs.
- If your scene use live streaming use the **VideoScreen** object as a display, you can inclose it freely, scale and move it. It is important to leave the name unchanged and the default material attached, called "Video"
- Textures and materials used exclusively in scene are stored in Textures, Materials folder and named: **XX_T_xxxxxxx, XX_M_xxxxxxx**
- Meshes you can name just as **XX_meshName**; **XX_AMP_name -** for Amplify Shaders; **XX_SG_name -** for Shader Graph; **XX_PS_name -** for Particle System (if it's included in prefab use **XX_PS** naming)

- Mesh assets should be exported as either .FBX or .OBJ
- Prefabs must have their position/rotation set to zero upon import, and should have their scale set to 1
- All Meshes must have a local pivot point positioned at the bottom center of the object, consistently in a corner of any modular objects, or where the object would logically pivot/rotate/animate
- Mesh prefabs need to be set up with a collider component that fits your mesh
- Content is being prepared for mobile/VR headsets and should have atlased textures in order to reduce performance impact.
- Assets should not have an excessive number of materials assigned to them
- Dimensions of textures should be pixel counts that are a power of 2 when appropriate
- Particle systems should be saved as a prefab.
- Skybox couldn't be changed.
- For lightmap bake you should use our XX_SampleScene settings for day and night, you can adjust intensity and other settings that affect appearance.
  (any changes in resolution/size/padding/non-directional are forbidden)
- We allow you to use only ***(exhibit/scene/camp)*** one/two/two 2048px lightmaps
- Delivered assets (without lightmaps) should be **NO MORE THAN 10MB**

- **Mesh which is less than 30k triangle**s, saved as a fbx should be less that 4 MB.
- How to check how much space would the textures occupy:
  When textures are imported to Unity they are stored in theirs orginal size.

In Inspector we can investigate how much space they occupy (screen here)
1. If imported textures are more that 2k, it's 'Max Size' should be set to 2048 or lower.
2. Compression should be set not higher than 'Normal Quality'
3. If the texture is too big (look at point 5. on screen) turn on the 'Use Crunch Compression'
4. Then adjust the Compressor Quality, hit apply and check texture's new size (5.)



Inspector

T_[redacted]_BaseMap Import Settings

Open

Texture Type: Default
Texture Shape: 2D
sRGB (Color Texture): ☑
Alpha Source: Input Texture Alpha
Alpha Is Transparency: □
Advanced
Non-Power of 2: ToNearest
Read/Write Enabled: □
Streaming Mipmaps: □
Generate Mip Maps: ☑
Border Mip Maps: □
Mip Map Filtering: Box
Mip Maps Preserve Cover: □
Fadeout Mip Maps: □
Wrap Mode: Repeat
Filter Mode: Bilinear
Aniso Level: 1

Default | iOS

Max Size: 2048
Resize Algorithm: Mitchell
Format: Automatic
Compression: Normal Quality
Use Crunch Compression: ☑
Compressor Quality: 64

Revert | Apply

T_[redacted]_BaseMap | RGB | R | G | B

2048x2048 RGB Compressed DXT1 sRGB 0.6 MB

AssetBundle: None | None

3. General requirements for 3d assets:

- Modular workflow where possible
- Base scale 1 unit = 1m
- Polycount: not more than 30k triangles.
- Maximum number of texture sets: 2-3

- Maximum texture resolution 2k pixels each
- Texture atlases are preferred or trim sheets (acceptable file formats: PNG, TGA)
- PBR textures for Unity - albedo, normal, metallic (or specular) and smoothness texture map
- You can work on the scenes provided by us - just rename it.

LODs - Depending on the number of polygons, you are required to perform LODs:

A - if you have 5K triangles or less
LOD0 -> original mesh
LOD1 -> Max 50% triangles of the entire asset

B - Between 5K and 15K triangles, you need to have 3 levels:
LOD0 -> original mesh
LOD1 -> Max 40% triangles of the entire asset
LOD2 -> Max 20% triangles of the entire asset

C - If you have between 15K to 30K triangles, you need to have 4 levels:

LOD0 -> original mesh
LOD1 -> Max 40% triangles of the entire asset
LOD2 -> Max 20% triangles of the entire asset
LOD3 -> Max 5% triangles of the entire asset

NOTE: (1) All models must have the correct UV maps and the correct texture as the original object. (2) Every LOD mesh should be a separate file, stored in Meshes folder.

UPDATED INSTRUCTIONS: The following language is added above as highlighted text

- When creating the scene, the scale of the "Shadow" object should be adjusted to the scale of the exhibit. If your stage has a lot of small objects spaced apart, it would be a good practice to create many "Shadow" objects to save space on the lightmap.
- Meshes you can name just as **XX_meshName**; **XX_AMP_name** - for Amplify Shaders; **XX_SG_name** - for Shader Graph; **XX_PS_name** - for Particle System (if it’s included in prefab use **XX_PS** naming)