

Creating Environment - Art Guidelines

Arena

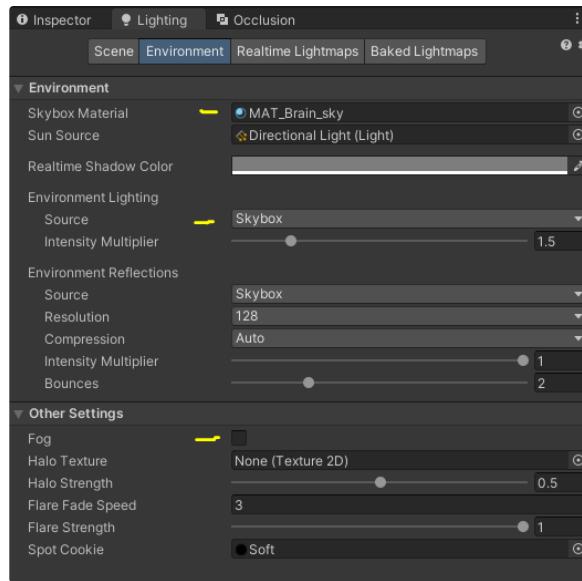
Keep the arena and stands as optimized as possible, simplify geometry in a distance and details/bevels in the arena rarely seen, the arena and backdrop must be well optimize to cater for gameplay and performance

- Most parts in the arena receive blood from particle collision and must use the ase_arena shader, the arenas are built out of modules where each module piece gets its own render texture for blood with its own material
- The backdrop (stands) do not support blood paint, use the shader ase_env_backdrop, this shader support a simple height fog that can be controlled globally
- The render texture used for blood paint is a small texture (128 pixels per arena piece)
- Each arena piece should be mapped 0-1 in UV0 with unique UV space for every part, no UV tiling will work with blood paint, try to keep UV islands together as much as possible, this will help the blood splattering look better.
- Make sure there are at least 16 pixels of padding between UV islands, since the blood painting extends it painting a bit to avoid seams. If the padding is to narrow there is a risk for leaking to another UV island
- Make sure door openings are large enough for the largest Rokibes and creature. Rokibes can be scaled up allot in custom game mode
- Every arena piece should have its own albedo and normal map textures, where the albedo also holds the Smoothness texture in the alpha channel. (Normal maps can be discarde)
- Vertex paint everything that is metallic with red, everything else should be black, enable “IsMetallic” on the material

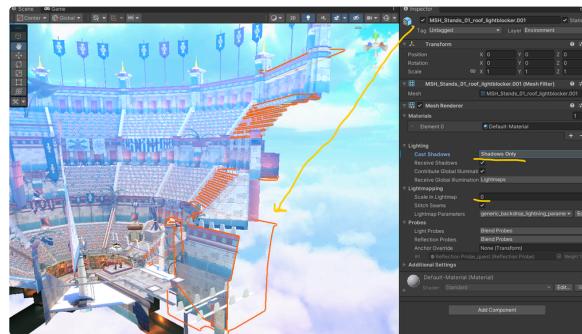


Lighting

- One mixed directional light in the scene (main light), this will be used to bake sun/moon light into lightmaps and also to lit dynamic object such as characters and weapons. All other lights should be set to baked
- In Lighting/Environment set the Skybox Material to the skybox you want to bake lighting from, also make sure you set Environment Lighting/Source to Skybox. this skybox will not render in runtime since we have a simple sky dome in the lighting prefab for this, also disable the standard fog



- If light baking non capped meshes results in light leakage or strange looking lightmaps, you can create a light blocker to resolve the issue, name it xxx_lightblocker, make sure its set to “Shadows Only” in the “Cast Shadows” dropdown, and “Scale in Lightmap” set to 0



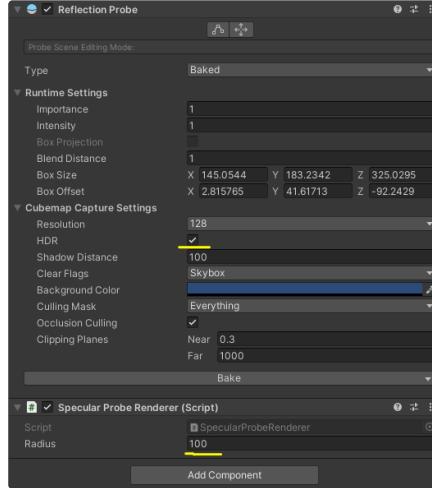
- The light probe group for an arena need to have more details near light sources and where light areas meet shadows, also since Rokibes and objects can be thrown out of the arena there need to be some light probes around the stands as well.



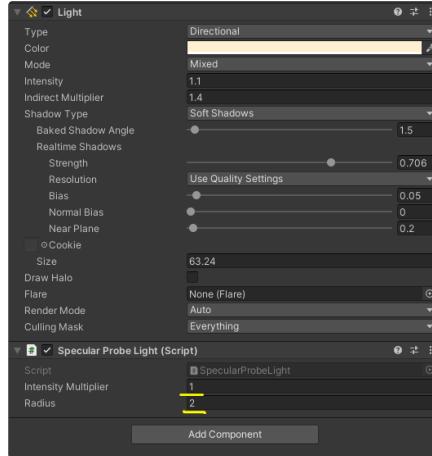
Specular from reflection probe

We use a plugin to get specular highlights from the main light (or any light source in the scene). This allows for baking highlights into the hdr reflection probe when scene lighting is baked [GitHub - zulubo/SpecularProbes: Bake specular highlights into Unity Reflection Probes, allowing baked lights to cast sharp specular highlights for free](#)

- Add the SpecularProbeRenderer script to the reflection probe in the lighting prefab, set the Radius and make sure you enable HDR on the reflection probe



- Add the SpecularProbeLight script on the light(s) and play around with the Intensity Multiplier and Radius until you are happy with the result (bake full scene to see the result)



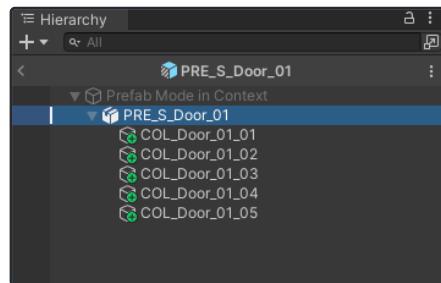
- You need to bake lighting on the full scene to see the result, re-baking only the reflection probe will not work for this, the highlight should look something like this.



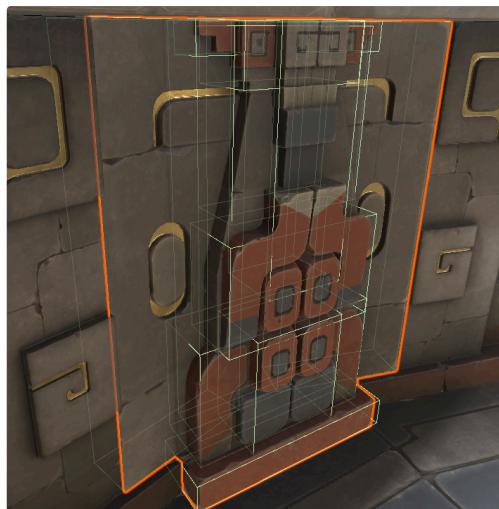
Arena Colliders [🔗](#)

Colliders need to be simple and built out of primitives (mainly boxes), colliders are also used for blood painting so they need to roughly match the render mesh. In some cases mesh colliders can be used as long as they are convex (check Convex on Mesh Collider)

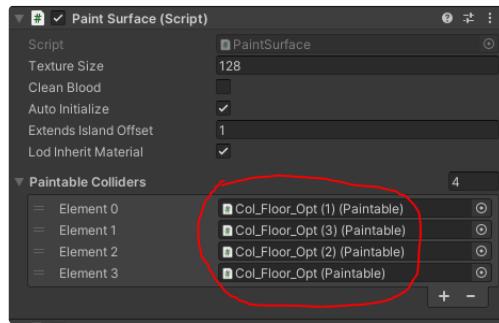
- Add colliders as children in the prefab of the render asset, follow naming convention (COL_same_as_mesh_01)



- Mind the detail level of the colliders, it needs to work with painting blood as well as serve as colliders, make sure it's well optimized but still working with blood paint, also make sure the colliders overlap each other



- Each arena part don't need its own colliders, on floors for example you can have one large or a few large colliders for the whole floor, then point the mesh renderer with the Paint Surface to those colliders instead of pointing the collider to the render mesh

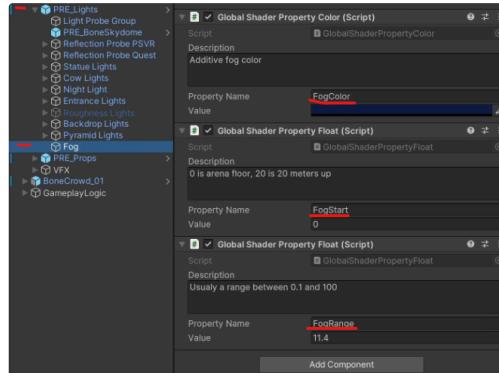


- Make sure the colliders are thick, should be at least 1 meter thick where possible

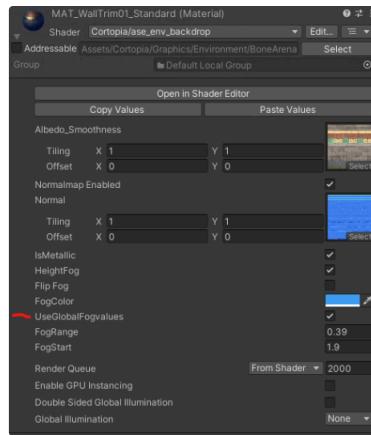
Global Fog

The simple height fog have global variables for the backdrop and crowd but can also be set per material for usage in corridors or rooms

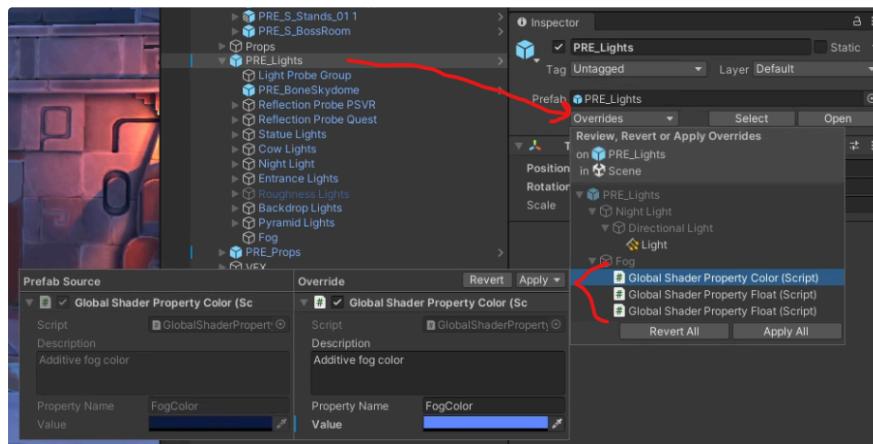
- In the lighting prefab of the level (must be a unique prefab per level), there should be a game object called Fog, this should hold the Global Shader Property scripts needed for the fog, its important that the names are exactly the same as the exposed global shader parameters (e.g FogColor, FogStart, FogRange)



- Make sure the materials that should have global fog have "UseGlobalFogValues" checked, this is set default when creating a new material with a shader supporting fog



- Once you have tweaked the fog in the scene, make sure you apply the overrides to the lightning prefab so that you don't submit the scene and only submits the lighting prefab



Blood splatter ↗

- Add a PaintSurface script to the game object holding the Mesh Renderer, and set texture size to 128
- On the colliders for the module piece add the script Paintable, drag the Game object holding the Mesh Renderer into the Paint Surface slot.