

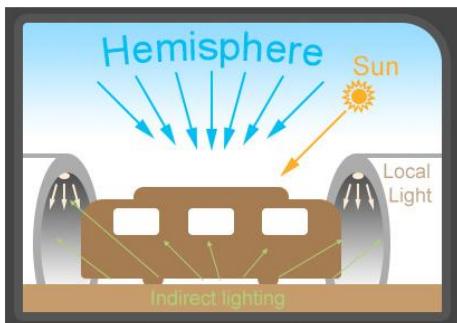


Lighting Strategy

Checked with version: 2017.3 - Difficulty: Intermediate

Before starting to create final assets and approaching lighting for a Scene, it is important to figure out your lighting strategy. At the start of a project, it is very easy for content creators - eager to start making cool things - to overlook this important step. Altering your lighting strategy late in development is a costly operation. Taking the time to get this right before you enter production will save time while giving better performance and a higher visual fidelity.

Much like anything in real life, there's almost always trade-off between the benefits and costs of one setup or another. Just as a Formula 1 Car isn't well suited for everyday groceries shopping compared to its gas saving hybrid car siblings. There are times however, where certain technologies will give you options to mitigate these trade-offs within specific constraints and knowing each features and trade-off will allow you to choose what's best for your project.



Going back to our Lighting, a typical Scene, at daytime, with outdoor areas can be broken down to 3 lighting components:

1. Hemisphere (Sky contribution).
2. Direct lights (Sun + Local lights).
3. Indirect lights (Bounced lighting).

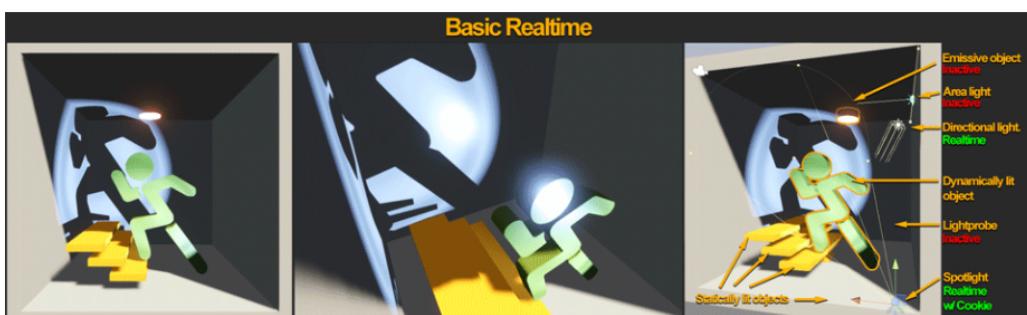
This seems like three simple components. How you choose to mix and match real time lights, mixed lights, baked lights, static objects and dynamic objects ends up creating a diverse range of potential lighting options.

In Unity we cater to lots of different lighting strategies and project scenarios.

[Find the documentation to understand lighting modes and setup here.](#)

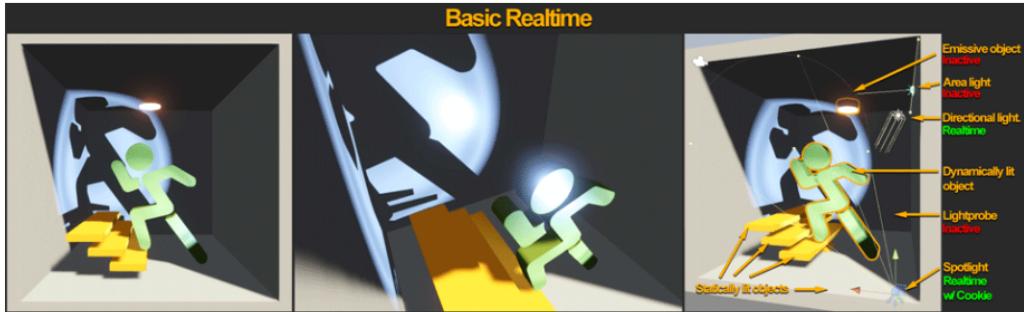
For newcomers, this can be overwhelming to figure out which setup and what the trade-off are for each setup. Let's distill this mass of information down to the most commonly used setup.

These 5 are the most commonly used lighting setup.



Visual notable differences between these options:

1. **Basic realtime**, the specular highlights from the light are visible but no indirect lighting.
2. **Baked**, soft baked shadows are visible, high resolution static indirect lighting, but there is no specular response from lights and dynamically lit object don't cast shadows.
3. **Mixed Lighting**, similar to Baked, but with specular response and dynamically lit object cast shadows.
4. **Realtime Light and GI**, proper indirect lighting response, specular response are visible, lights are all moveable and updateable, but there's no angular soft shadow.
5. **Guns Blazing all options enabled**, depending on the settings of each light you can achieve the combination of all the above options.



The slideshow above showcase baked Ambient Occlusion when enabled. NOTE: Realtime GI can't bake static ambient occlusion and hence not included.

Here are the general characteristics for each configuration:

Basic Realtime lighting + Ambient (with no Realtime GI or Baked GI).



Typical platform target: Console and PC. Generally used in stylistic visual project and prototype phase.

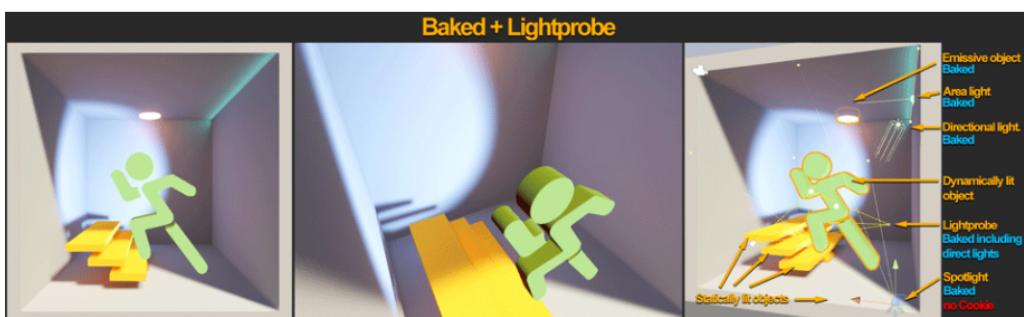
Advantage:

- All direct lights and shadows are real-time, therefore movable.
- Fast iteration because of no precompute or baking and mesh preparation.
- Dynamic objects and Static objects are lit using the same method, no Light Probes required.

Disadvantage:

- No Hemisphere occlusion, just straight skybox/ambient value and color in area not lit by direct lighting.
- Without "Global Illumination" / indirect lighting component, scene might not give the best visual outcome.

All baked lighting + Light Probe.



Typical platform target: Mobile platform, VR, console and low end PC. Generally used in games where runtime performance is an issue but there's room in memory, such as top down isometric mobile games and high frame rates VR game.

Advantage:

- All lights are baked for static objects, produce ambient occlusion and indirect lighting.
- Area light bake support and soft shadow angle can be baked onto statically lit objects.
- Fastest in runtime performance among the options listed here.

Disadvantage:

- Can slow down lighting iteration since lights are baked, therefore requiring light rebuild on Scene changes if progressive Lightmapper aren't used.
- Dynamically lit objects are lit using Light Probes only.
- No Specular highlights from light source, relying only on cubemap/reflection.
- No shadowing from dynamic objects.
- Can cost a lot of runtime memory depending on how much lightmap textures used in the scene.
- Might require authoring texture coordinates channel 2 (UV2 for lightmap).

Mixed lighting with Shadowmask + Light Probe.



Typical platform target: VR, console and PC. Generally used in majority of console games and pc games where time of day lighting such as sun movement is not important. Advantage:

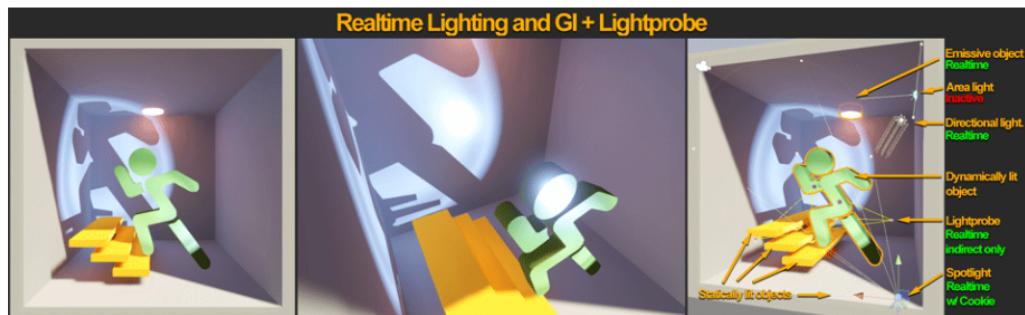
- Similar to all baked lighting, but in Mixed lighting, Dynamic object gets real-time specular and cast realtime shadows, while the static object gets baked shadowmask resulting in better visual quality.

Disadvantage:

- Object can only have 4 shadowmask limit.
- Cost additional performance in runtime for rendering the real-time lights.
- Care needs to be taken in using mixed lights as they can drastically affect performance in certain setups.

The above list is an oversimplification description of shadowmask lighting. [Complete information can be found here.](#)

Realtime lighting with Realtime GI + Light Probe.



Typical platform target: Console and PC. Generally used in open area game where time of day lighting updates is required and dynamic lighting effect are required as part of the game design. Advantage:

- This allow fast lighting iteration with realtime indirect lighting.
- Dynamic and static objects get real time specular and shadows.
- Can use less memory than baked lighting for giving indirect lighting effect.
- Fixed cost CPU time performance for updating GI.

Disadvantage:

- Occlusion isn't as detailed as baked lighting and usually must be augmented by Screen Space Ambient Occlusion (SSAO) and per object texture baked AO.
- No area/ light angle soft shadows for static objects.

- Care needs to be taken in using realtime lights as they can drastically affect performance in certain setups.
- Precompute times (Generate lighting) time can take significant amount of time if there's too much object contributing to the static lighting especially without optimized UV setup (Requires authoring texture coordinates channel 3 (UV3) for Enlighten Realtime GI further optimization and/or projection fixes.)

In depth information in optimizing Realtime GI can be found [here](#).

Guns blazing, all option enabled.



Typical platform target: Console and PC. Generally used for games with high fidelity requirements with tightly controlled memory usage and performance limit. Best enabled when the content creators completely understood each individual system and has proper knowledge on handling each of the lighting combination implication.

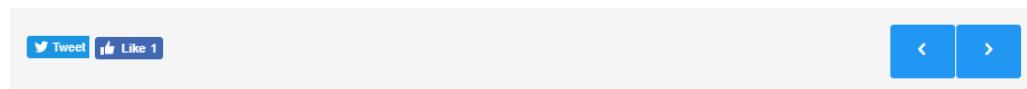
Advantage:

- This is the complete set of lighting features, it gives your content creator all the functionality.

Disadvantage:

- Can potentially burn performance at run time, memory usage.
- Increase the workflow burden (UV authoring and baking time).

For faster iteration and learning lighting a Scene, responsive visual feedback is necessary. For this reason, the Spotlight Tunnel Sample Scene is using Realtime Lighting w/ Realtime GI. This will give us a nice range of specular response, good bounce lighting, and let us change our lights on the fly.



 unity ↑

Purchase	Download	Resources	About Unity	Get Unity news
Subscription	Unity	Learn	Blog	<input type="text"/> Enter your email here... Sign up
Asset Store	Beta Program	Community	Events	
Unity Gear	Web Player	Documentation	Careers	
Resellers	Press material	Unity QA	Contact	
	Whitepapers	FAQ	Press	
Education		Services Status	Partners	
Students	Unity Labs	Certification	Affiliates	
Educators	Labs	Connect	Security	
Center of Excellence	Publications			

Partners

 Cloud Moolah
 facebook
 Google
 Intel
 Microsoft
 Nintendo
 oculus
 SONY
 vuforia
 mii

© 2019 Unity Technologies [Legal](#) [Privacy Policy](#) [Cookies](#)

[!\[\]\(462ce7f456dedf947a62ec467a11aa02_img.jpg\)](#)
[!\[\]\(0584d3fbef68418aeee6331d06e115b9_img.jpg\)](#)
[!\[\]\(338d4c03545dc3a3eeee9b04d1285ce0_img.jpg\)](#)
[!\[\]\(c3a963ffff54e9d700faa095d5a097ea_img.jpg\)](#)
[!\[\]\(6838ac7dbb723befb98be5f569d79c44_img.jpg\)](#)

"Unity", Unity logos, and other Unity trademarks are trademarks or registered trademarks of Unity Technologies or its affiliates in the U.S. and elsewhere (more info [here](#)). Other names or brands are trademarks of their respective owners.