

Sommario

INTRODUZIONE.....	5
Illuminazione e Shading in Unity 5.....	7
Il rendering: interazione tra luce e superfici.....	7
L'illuminazione in Unity 5: ambientale, diretta, indiretta; Global Illumination.....	8
In pratica: la scena che utilizzeremo nei nostri esempi.....	12
Proprietà generali: Project Settings, Rendering Paths, Lighting.....	16
Impostazioni del progetto: Quality e Player Settings.....	16
Scelta del Rendering Path.....	17
Metodi Rendering Path: Forward vs Deferred.....	18
Scelta del Color Space.....	20
In pratica: le impostazioni del progetto utilizzato per gli esempi di questa guida.....	21
L'illuminazione ambientale.....	23
Cos'è l'illuminazione ambientale.....	23
Parametri di base: Source, Intensity, GI.....	24
In pratica: esempio di illuminazione ambientale nella nostra scena	26
Lo Skybox.....	30
Cos'è uno Skybox e come si crea un Material Skybox.....	30
6-Faces, Cubemap, Procedural; realizzare le immagini (plugin).....	31
Assegnare lo Skybox alla scena corrente.....	34
Distinzione tra Skybox di telecamera (Clear Flag) e di scena.....	34
In pratica: creare uno Skybox (tramite plugin) ed assegnarlo alla scena corrente.....	34
Fonti di luce standard: Directional.....	39
Parametri di base (comuni anche ad altre luci): colore, intensità, Bounce Intensity.....	39

Ombre Hard e Soft; Bias e Normal Bias.....	40
Baking (cenni introduttivi).....	43
Directional come Sun in Ambient Light; orari della giornata; il Sole di una Texture Skybox.....	43
In pratica: utilizzare una Directional Light (e Sun) nella scena.....	45
Fonti di luce standard: Point Light.....	49
Parametri: Range, intensità, colore e ombre.....	49
In pratica: utilizzare una Point Light nella scena; Light Leaks.....	51
Fonti di luce standard: Spot Light.....	55
Parametri di una Spot Light.....	55
In pratica: utilizzare una Spot Light nella scena.....	56
Introduzione all'illuminazione globale statica e real-time in Unity 5.....	59
Illuminazione indiretta e Global Illumination.....	59
Illuminazione indiretta statica: Baked e Precomputed.....	59
Illuminazione indiretta dinamica: Light Probes, Reflection Probes.....	60
Impostazioni delle Lightmaps (Baked) e Lightmap Parameters.....	61
In pratica: impostazioni Lightmap Parameters.....	62
Baked e Precomputed GI.....	67
Enlighten, Systems e precalcolo.....	67
Baked e Precomputed Real-Time; impostazioni in Lighting.....	70
Il Baking e le Lightmaps.....	72
Lightmapping Bake nelle versioni precedenti e in Unity 5.....	73
In pratica: visualizzare i Cluster (e non solo) nell'editor.....	73
La fonte di luce Area Light e il suo baking.....	78
Baked (Static).....	78
In pratica: Baking di una fonte di luce Area Light nella scena d'esempio.....	79
Illuminazione indiretta	

con Precomputed Realtime.....	82
Precomputed Realtime GI in Unity 5.....	82
In pratica: illuminare una scena con Precomputed Real-Time.....	83
 Luce ambientale in Real Time.	
Reflection Source.....	89
Luce ambientale in GI.....	89
Reflection Source.....	90
In pratica: esempio pratico di luce ambientale in GI nel nostro progetto.....	91
 Light Probes, Reflection Probes: illuminazione e shading su oggetti dinamici.....	94
A cosa servono le Light Probes.....	94
Creare e disporre le Light Probes: Light Probes Group.....	95
Baking delle Light Probes.....	96
In pratica: esempio di utilizzo delle Light Probes nel nostro progetto	97
Riflettere l'ambiente su oggetti statici e dinamici: Reflection Probes	101
In pratica: esempio di utilizzo delle Reflection Probes nel nostro progetto.....	102
 Materiali emissivi e real-time GI.....	105
Emissive Materials e Lightmap Static.....	105
Inspector Static e problemi a runtime con Emissive Non Static....	106
Le Light Probes e gli oggetti dinamici da illuminare tramite Emissive.....	106
Oggetti emissivi in movimento.....	107
In pratica: un materiale emissivo Real-Time animato, con script, nel nostro progetto.....	107
 Camera HDR e Post-Processing Stack.....	111
LDR vs HDR; Tonemapping.....	111
HDR in Unity; HDR per la telecamera.....	112

Buffer e filtri HDR; note su Forward e Deferred Rendering.....	113
Il package Post-Processing Stack e gli effetti per le telecamere....	114
In pratica: utilizzare il pacchetto Post-Processing Stack.....	115
In pratica: Bloom e altri effetti.....	118

INTRODUZIONE

Questa è una guida introduttiva ai sistemi di illuminazione diretta, indiretta e Global Illumination in Unity 5, rivolta a chi desidera conoscere le caratteristiche fondamentali delle fonti di luce e del sistema di illuminazione globale, sia precalcolato (Baked) che in tempo reale (Precomputed).

La guida è stata realizzata con la versione 5.6 di Unity ed è indirizzata a chi ha una conoscenza di base del programma e desidera approfondire la conoscenza dei sistemi di illuminazione, in modo da migliorare la propria padronanza di tale aspetto e, quindi, il risultato visivo nei propri progetti.

Le tecniche e gli strumenti verranno mostrati con esempi, in modo da mettere in pratica le informazioni riportate nei vari capitoli; verranno quindi trattate, tra le altre cose: le fonti di luce standard e i materiali emissivi; la definizione dell'illuminazione ambientale e dello Skybox; le caratteristiche dei sistemi di illuminazione Baked e Precomputed Realtime per la Global Illumination (diretta ed indiretta); l'utilizzo delle Light Probes e delle Reflection Probes per l'illuminazione e l'ombreggiatura in tempo reale degli oggetti dinamici; il pacchetto Post-Processing Stack, per realizzare effetti visivi in post-produzione.

Le parti pratiche dei capitoli della guida sono disponibili gratuitamente sotto forma di videotutorial pubblicati sul canale Youtube dell'autore, per un totale di poco più di un'ora di contenuti distribuiti in 17 video.

Dal sito web dell'autore sarà inoltre possibile scaricare, gratuitamente e senza registrazione, il pacchetto Asset, contenente appunto il package Unity con la scena utilizzata negli esempi, per cui scaricandolo sarà possibile ricreare e modificare tali esempi; inoltre, nel pacchetto sono presenti anche le immagini a colori della guida, utili se si legge l'eBook su un dispositivo Kindle con schermo in bianco e nero.

Qui di seguito, il link alla pagina di questa guida sul sito web dell'autore, dove troverete i link ai videotutorial gratuiti e al download del pacchetto

Asset (package unity e immagini a colori dell'eBook):

<https://www.francescomilanese.com/it/virtuemart-tutorials-ita/unity-ita/unity-5-illuminazione-ebook-videocorso.html>

(in alternativa, potete raggiungere la pagina dal menù **Tutorials** sul sito web dell'autore, all'indirizzo <https://www.francescomilanese.com/it/>).

Francesco Milanese è un Blender Foundation Certified Trainer dal 2010, modellatore 3D e autore di ebook, videocorsi e seminari su Blender 3D e altri software di CG 2D / 3D e Programmazione.

Laureato in Informatica Specialistica presso l'Università degli Studi di Catania, si occupa di Computer Grafica 3D dall'adolescenza. Ha realizzato e gestito portali web come *redbaron85.com* e *effe12.com*, prima di passare definitivamente a *francescomilanese.com*, pubblicando periodicamente videotutorials, ebook e guide sull'utilizzo di Blender 3D e altri software di Computer Grafica e Programmazione.

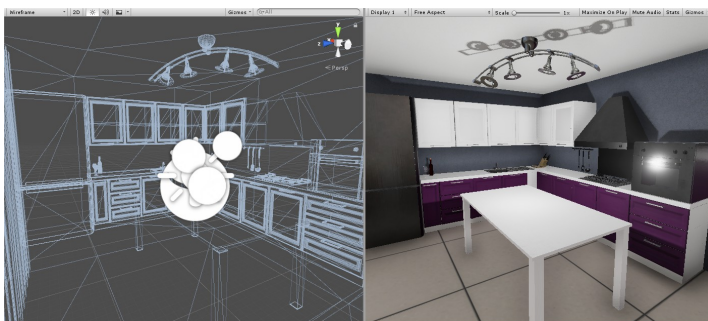
CAPITOLO 1

Illuminazione e Shading in Unity 5

Il rendering: interazione tra luce e superfici

Quando un software di grafica 3D renderizza una scena tiene conto delle fonti e dei sistemi di illuminazione presenti in essa, della natura delle superfici illuminate e, quindi, dell'interazione tra luci e superfici, eventualmente aggiungendo anche degli effetti, detti di Post-Produzione, al risultato; in effetti, il rendering è il prodotto di questa operazione: l'elaborazione del colore di ogni punto di ogni superficie inquadrata dalla telecamera virtuale e l'applicazione degli effetti speciali.

A seconda che il risultato da ottenere sia un'immagine o un'animazione (un risultato statico, nel senso che può essere solamente visualizzato senza possibilità di modifica in tempo reale) o una visualizzazione in tempo reale, dinamica, verranno utilizzate tecniche e approssimazioni diverse per far fronte alle diverse esigenze. Unity 5 è un motore di gioco per applicazioni che possono essere eseguite su una gran varietà di piattaforme di destinazione (dai dispositivi mobile ai sistemi desktop di fascia alta, passando per vari tipi di console), quindi il rendering viene realizzato tenendo conto delle richieste specifiche di un sistema in tempo reale e alcuni effetti complessi vengono approssimati o eliminati del tutto per favorire la velocità d'esecuzione. Ciò riguarda, ovviamente, anche alcuni effetti di illuminazione e ombreggiatura di oggetti e scene.



Il rendering è il risultato dell'elaborazione dell'illuminazione in un ambiente virtuale: le geometrie e le loro interazioni con fonti di luce, inquadrare dalla telecamera virtuale (a sinistra), producono l'immagine finale (a destra).

È necessario quindi parlare dei vari tipi e dei parametri dei sistemi di illuminazione presenti in Unity 5, dei tipi di superfici e dei principali tipi di Textures che ci aiutano a fornire un aspetto visivo o a definire determinate proprietà fisiche (quindi, ancora una volta, di interazione con la luce) per le superfici alle quali sono applicate; in questa guida, ci occuperemo del primo punto: l'illuminazione.

L'illuminazione in Unity 5: ambientale, diretta, indiretta; Global Illumination

In Unity 5 possiamo individuare tre principali metodi di illuminazione: ambientale, diretta e indiretta.

L'illuminazione ambientale illumina in maniera uniforme tutte le superfici, senza proiettare ombre su altri elementi e, a meno di non aggiungere effetti particolari, senza tenere conto di occlusioni o altri aspetti che potrebbero modificare l'ombreggiatura degli oggetti.

Può essere utilizzata per produrre un risultato in stile cartoon, per dare una luminosità di base a tutti gli elementi (specialmente in scene al chiuso) o in fase di test delle scene, quando ancora si stanno posizionando gli elementi e si iniziano a provare colori e textures sulle superfici, senza introdurre le fonti di luce e sistemi di illuminazione più complessi.



L'illuminazione ambientale (non globale) produce un'illuminazione piatta, uniforme, sulle superfici degli oggetti.

L'illuminazione diretta è quella prodotta dalle fonti di luce standard (luci omnidirezionali, direzionali, faretti spot) che illuminano gli oggetti che si trovano nel loro raggio d'azione; quando i raggi luminosi emessi da queste fonti di luce impattano su un oggetto, producono delle ombre sugli altri oggetti.



L'illuminazione diretta produce ombre ma non illuminazione di rimbalzo; gli oggetti non direttamente illuminati, in questa immagine, sono visibili per via dei riflessi ambientali, presenti di default in Unity.

L'illuminazione diretta è quindi quella che consente di illuminare gli oggetti in tempo reale e di proiettare le loro ombre sulle superfici, ma non calcola l'illuminazione di rimbalzo.

Con Unity 5 è stato introdotto un sistema di illuminazione indiretta che tiene conto anche dell'illuminazione indiretta in real-time, ovvero della luce di rimbalzo. Come vedremo nel corso di questa guida, l'illuminazione indiretta in Unity può essere calcolata con due diversi metodi, a seconda delle caratteristiche dell'oggetto; inoltre, bisognerà ricorrere a metodi diversi per oggetti statici (che non varieranno in posizione e altre caratteristiche nella scena, per cui la loro illuminazione potrà essere precalcolata) e dinamici (ad esempio, personaggi o altri oggetti trasformabili).



Una scena illuminata con Global Illumination: sono presenti le ombre date dall'illuminazione diretta, la luce di rimbalzo (indiretta) e i riflessi ambientali.

L'unione dei sistemi di illuminazione diretta ed indiretta consente di implementare la Global Illumination in Unity, così da poter ottenere risultati più realistici nell'illuminazione delle scene, anche in tempo reale.

Prima di concludere questo capitolo introduttivo, va detto che alle fonti di luce standard e all'illuminazione ambientale si aggiunge un terzo tipo di strumenti di illuminazione: quello dei materiali emissivi, che consentono di rendere alcune geometrie dei veri e propri emettitori luminosi, eventualmente mediante l'utilizzo di Textures mappate sulle geometrie per definire colore e intensità dell'emissione luminosa sulle varie parti dell'oggetto.



La scena illuminata mediante un materiale Emissive posto in prossimità della cappa

Per quanto riguarda, invece, le tecniche di illuminazione, Unity fa ricorso sia a metodi statici (Static Lightmaps) e dinamici (Light Probes), a seconda delle scene e degli oggetti da illuminare, per implementare la Global Illumination. Descriveremo ed utilizzeremo tutti questi elementi nei prossimi capitoli.

In pratica: la scena che utilizzeremo nei nostri esempi

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

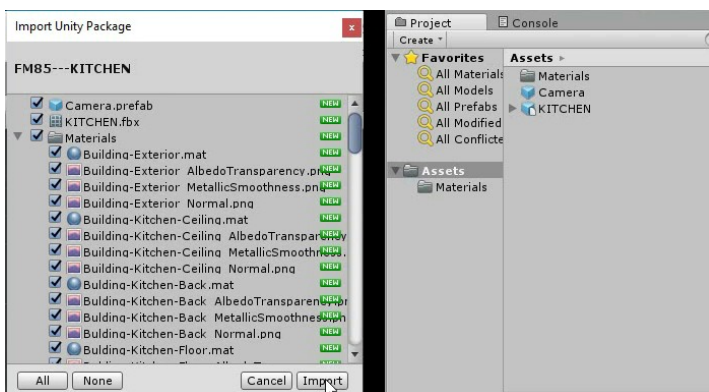
La scena che utilizzeremo nei nostri esempi è quella di una cucina, con la telecamera virtuale posizionata all'interno dell'ambiente. Non si tratta di un'ambientazione particolarmente accattivante, ma alcune sue caratteristiche ci consentono di analizzare le varie fonti di luce e i

diversi sistemi di illuminazione presenti in Unity 5; in particolare:

- l'ambiente è completamente chiuso, ma sono presenti delle aperture semitrasparenti, per cui è possibile studiare l'illuminazione proveniente dal Sole (Directional Light) e dallo Skybox;
- è possibile inserire tutte le altre fonti di luce standard singolarmente e insieme, per valutarne gli effetti: le Spot Light nel lampadario, un'Area Lamp nella cappa, alcune Point Light per le spie del forno o all'interno dello stesso;
- il tavolo può essere utilizzato per analizzare le ombre proiettate dalle varie fonti di luce;
- per tutti gli oggetti è possibile realizzare un'illuminazione precalcolata (e vedremo come farlo, generando le varie Lightmap), ma sono presenti anche oggetti che possono essere spostati, così da valutare gli effetti di illuminazione e ombreggiatura, inclusi i riflessi speculari, in tempo reale;
- sono presenti superfici semitrasparenti e metalliche, per cui sarà possibile osservare gli effetti di illuminazione e ombreggiatura anche su questi oggetti, oltre che sui non metalli;
- è possibile valutare alcuni effetti speciali di post-produzione come Bloom, Depth of Field ed altri.

Chi acquista il pacchetto Assets di questo ebook troverà, all'interno dell'archivio, il Package per Unity 5 contenente le geometrie, i materiali texturizzati e la telecamera virtuale utilizzati negli esempi di questa guida, per cui potrà effettivamente ricreare gli esempi qui discussi o modificare l'ambiente virtuale (sostituendo o spostando oggetti, punto di vista della telecamera, fonti di luce, eccetera) per sperimentare personalmente con un ambiente virtuale già pronto all'uso.

14 L'illuminazione in Unity 5 – Guida introduttiva



Importare il pacchetto .unitypackage dal menù Asset - Import Package - Custom Package (figura a sinistra), oppure trascinandolo da disco in Project; quando gli elementi appariranno in Project (figura a destra), trascinarli Camera e KITCHEN in Hierarchy per inserirli nella scena.

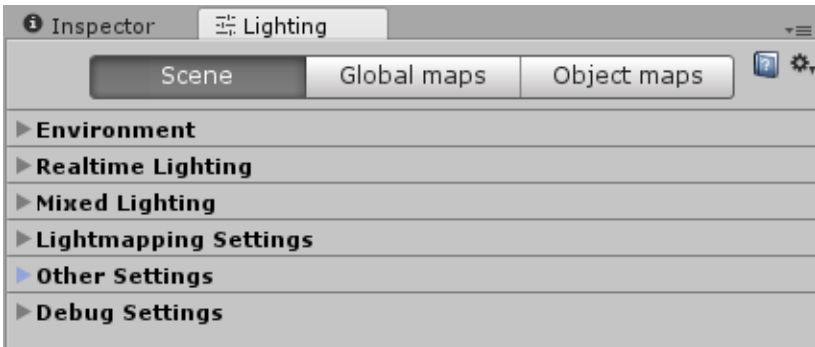


La scena che utilizzeremo, qui vista dalla telecamera virtuale, nella finestra Game.

Le impostazioni generali di illuminazione del progetto sono l'argomento principale di questa guida, per cui è importante averle a portata di mano sotto forma di scheda ancorata alla finestra principale di Unity; come prima operazione da effettuare prima di chiudere questo capitolo,

quindi, apriamo una scheda Lighting e posizioniamola accanto all'Inspector, in questo modo:

- dal menù Window, selezioniamo Lighting – Setting;
- clicchiamo sulla tab Lighting della finestra floatante che apparirà a video e, tenendo premuto il tasto sinistro del mouse, trasciniamo tale etichetta accanto a quella dell'Inspector.



La scheda Lighting, richiamabile dal menù Window, qui posta accanto all'Inspector e con tutte le sotto-sezioni chiuse.

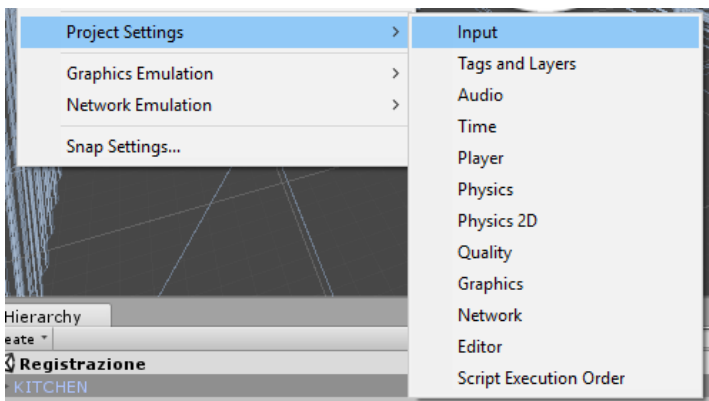
CAPITOLO 2

Proprietà generali: Project Settings, Rendering Paths, Lighting

Impostazioni del progetto: Quality e Player Settings

Quando si crea un nuovo progetto è bene impostare sin da subito alcune caratteristiche dell'interfaccia, dei controlli, dei dispositivi di destinazione e della resa visiva; in questa sede, dovendo trattare illuminazione e rendering, ci concentreremo su quest'ultimo punto.

Le impostazioni globali del progetto si trovano in Edit – Project Settings e sono suddivise per ambiti.



Si accede alle impostazioni generali del progetto da Project Settings, nel menù Edit.

Nella sezione Quality possiamo impostare, intuitivamente, il grado di qualità visiva desiderato, scegliendo tra i Presets (da Fastest a Fantastic) e specificando, per tali Preset o per configurazioni personalizzate (da creare con “Add Quality Level”) parametri come la qualità del filtro anti-scalettatura Anti-Aliasing, l'utilizzo dei Reflection Probes per l'elaborazione dei riflessi luminosi sugli oggetti, il numero di fonti di luce da trattare come per-pixel, ed altro ancora.

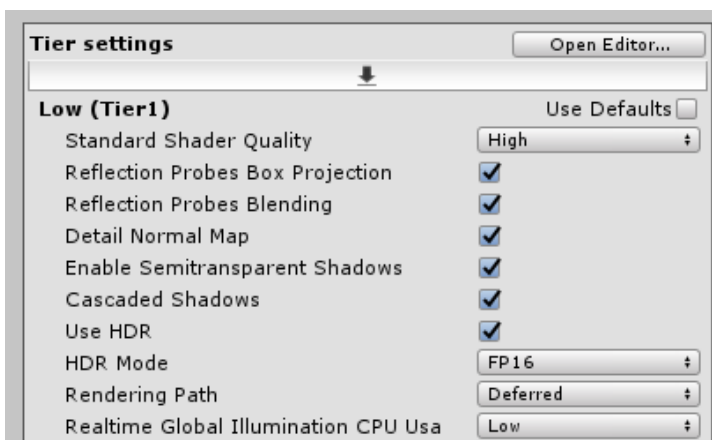
Elencare ora tutti i parametri avrebbe poco senso, ma nel corso della guida tratteremo buona parte delle impostazioni qui presenti e, a quel punto, si potrà tornare a tale scheda per personalizzare, se necessario, alcuni elementi.

Scegliendo invece, sempre dal menù Edit – Project Settings, la voce Graphics, apriremo tale scheda nell'Inspector, dove potremo specificare alcune impostazioni grafiche per il progetto tenendo conto, tra l'altro, dell'hardware di destinazione, differenziando le impostazioni (e, quindi, la qualità) in base a questo parametro. Tra i vari parametri presenti in Player Settings esaminiamo, in particolare, Rendering Path.

Scelta del Rendering Path

I Rendering Path sono metodi di rendering, ciascuno dei quali ha le sue caratteristiche ed è indicato per determinate piattaforme di destinazione o tipi di giochi e applicazioni. Unity supporta vari tipi di Rendering Path ma la scelta cade, generalmente, su Forward o Deferred; le principali differenze tra questi due metodi verranno esposte a breve.

La scelta del Rendering Path va effettuata, a livello di progetto, nella scheda Graphics di Edit – Project Settings, ma è possibile specificare un metodo in maniera esplicita per ciascuna telecamera virtuale, utilizzando anche in questo caso un campo dal nome “Rendering Path”, nell'Inspector di ogni telecamera, per cui l'impostazione propria di ciascuna telecamera sovrascriverà le impostazioni generali del progetto.



Scelta del Rendering Path per il progetto in Edit - Project Settings – Graphics.

Unity valuterà comunque le caratteristiche hardware e software del dispositivo di esecuzione e, in caso di risorse non adeguate (poca memoria video o assenza di supporto per determinati Shader), passerà automaticamente al Rendering Path di livello inferiore (tipicamente, da Deferred a Forward).

Metodi Rendering Path: Forward vs Deferred

Il Deferred Shading è la modalità di rendering (Rendering Path o anche pipeline di rendering) che garantisce la miglior qualità in termini di illuminazione, proiezione di ombre ed altri effetti, perché gli effetti di ombreggiatura vengono calcolati per pixel, per cui produce ottimi risultati anche con più fonti di luce in tempo reale ma, proprio per questi motivi, richiede un hardware piuttosto potente. Non presenta il supporto all'Anti-Aliasing e non renderizza gli oggetti semitrasparenti (che vengono renderizzati ricorrendo automaticamente al Forward Rendering).

Il Forward Rendering supporta le caratteristiche base di Unity ed è più veloce del Deferred Shading (a discapito della precisione, in alcuni contesti), perché solo un certo numero di fonti di luce real-time, specificato nel parametro “Pixel Light Count” della scheda Project Settings – Quality, viene considerato per-pixel nell'ombreggiatura della scena, mentre le altre vengono calcolate per-vertex (interpolazione sulle superfici) e altre ancora con un'approssimazione per-object (metodo HS). Forward Rendering è supportato da tutti i sistemi (da mobile a desktop passando per console), mentre per Deferred sono necessari, ad esempio, lo Shader Model 3 su sistemi Desktop e OpenGL ES 3 su sistemi mobile.

La distinzione più importante riguarda, nello specifico, l'illuminazione: nel Deferred Shading non vi è limite TEORICO (ma, in pratica, dipende dall'hardware) alla quantità di fonti di luce da considerare per-pixel e, quindi, alla qualità e alla fedeltà della resa visiva; di contro, il Forward Rendering è più veloce ma più approssimativo, perché renderizza un oggetto in uno o più passi considerando le fonti di luce per oggetto in questo modo: un ristretto numero di fonti di luce più intense o vicine vengono considerate per-pixel; altre fonti di luce vengono considerate per-vertex; tutte le altre vengono calcolate con l'approssimazione Spherical Harmonics. Ci sono delle sovrapposizioni, ovvero l'utilizzo di tecniche miste per le fonti di luce di confine tra un metodo e un altro, per evitare un effetto “salto” nell'illuminazione, ma è chiaro che la fedeltà è garantita solo per un numero limitato di fonti luminose.

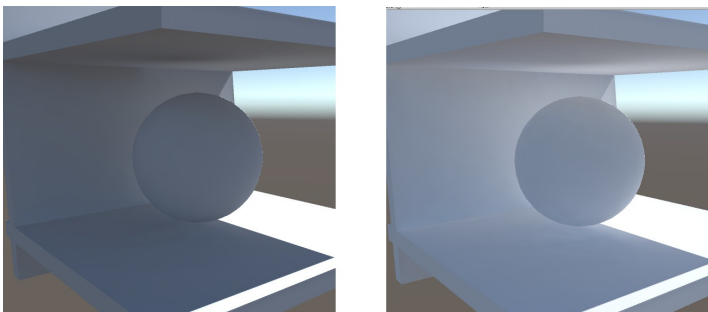
Di default, la modalità di rendering selezionata a livello globale è Forward, ma per lo studio dell'illuminazione e per le prime prove si consiglia di passare a Deferred, a livello di progetto (e, chiaramente, anche per le singole telecamere); Unity si affiderà automaticamente a Forward per alcuni aspetti non gestiti da Deferred, come ad esempio i materiali semitrasparenti.

Scelta del Color Space

Nella scheda Player di Project Settings troviamo invece il parametro Color Space, nella sezione Other Settings della scheda, per le varie piattaforme di destinazione.

Senza scendere troppo in dettaglio, possiamo dire che tale parametro definisce il modo in cui Unity effettuerà, dietro le quinte, i calcoli dell'illuminazione della scena; in particolare, la modalità Linear è quella più corretta dal punto di vista matematico, mentre la modalità Gamma è basata sulla percezione che i nostri occhi hanno delle variazioni di illuminazione.

Dal punto di vista pratico, lavorando in Linear Color Space i contorni delle zone illuminate saranno meglio definiti (più netti, rispetto a quanto visibile in Gamma Color Space). La differenza più evidente si ha, comunque, nella gestione delle illuminazioni particolarmente intense: con Gamma, l'illuminazione sulle parti texturizzate potrebbe risultare troppo intensa, per via della correzione Gamma già applicata alle Textures che, sommandosi all'illuminazione della scena, farà apparire le superfici più chiare di quanto dovrebbero essere.



Variazioni della luminosità in una semplice scena cambiando solo il Color Space: a sinistra Gamma, a destra Linear.

È consigliabile impostare il Color Space subito dopo la creazione di un nuovo progetto, prima di effettuare qualsiasi altra operazione, in modo da iniziare subito con la modalità desiderata. Questa raccomandazione può rivelarsi utile anche per verificare la compatibilità con il sistema in uso o la piattaforma di destinazione (da impostare nella scheda Build Settings), perché non tutti i sistemi hardware o software supportano l'uno o l'altro sistema.

Il Color Space consigliato è Linear ma, come detto precedentemente, a volte non è proprio possibile utilizzare tale modalità, a seconda del sistema in uso e/o di quello di destinazione (ad esempio, WebGL), per cui in effetti è bene informarsi sulle potenzialità dei dispositivi di destinazione, in particolar modo se si desidera effettuare il deploy per dispositivi mobile. Gli esempi mostrati in questa trattazione sono stati realizzati in Linear Color Space con impostato, come dispositivo di destinazione, un sistema Desktop (PC, Linux & Mac Standalone).

In pratica: le impostazioni del progetto utilizzato per gli esempi di questa guida

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

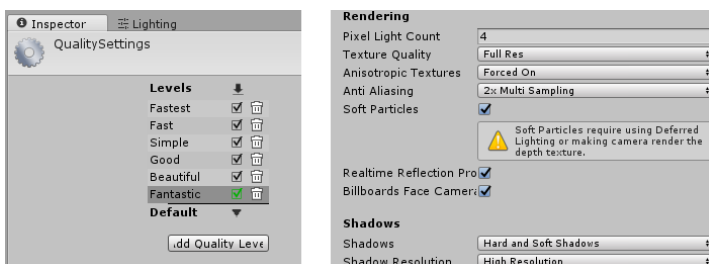
Vediamo quindi le impostazioni da utilizzare per la scena d'esempio, fornita nel pacchetto Assets, le cui immagini verranno mostrate nei prossimi capitoli della guida.

Per quanto riguarda le impostazioni di qualità in Edit – Project Settings – Quality, sto scegliendo il preset Fantastic che, tra i preset, è quello con la qualità visiva migliore; va detto, comunque, che un paio di impostazioni non presentano i valori massimi possibili nemmeno con questo preset:

- l'Anti-Aliasing è attivo, in modalità 2x; è possibile passare

anche alla modalità 4x o 8x tramite l'apposito selettore, se lo si ritiene necessario (in presenza di molti oggetti dai bordi dritti, ad esempio, visto che il filtro si occupa proprio dell'anti-scalettatura);

- la qualità delle ombre è impostata su High Resolution, ma il selettore mette a disposizione l'opzione Very High Resolution.



Le impostazioni Quality per il nostro progetto, con il preset Fantastic selezionato.

Sto immaginando un progetto da destinare a dispositivi Desktop (Win, Linux o Mac), per cui, oltre alle impostazioni di qualità Fantastic in Quality, posso impostare, nella scheda Project Settings – Player, anche la modalità Color Space su Linear, che è quella che implementa operazioni corrette dal punto di vista matematico e che restituisce risultati migliori, più realistici.

Per quanto riguarda, infine, le impostazioni del gruppo Graphics di Project Settings, sto impostando l'utilizzo del metodo Deferred in tutti e tre i tiers di destinazione del progetto; si tratta del metodo che fornisce la qualità migliore per l'illuminazione e le ombre, in particolare in presenza di fonti di luce real-time.

CAPITOLO 3

L'illuminazione ambientale

Cos'è l'illuminazione ambientale

Quella ambientale (Environment) è una tecnica di illuminazione che consiste nel fornire una luminosità di base a tutte le superfici di tutti gli oggetti, senza calcolare ombre o altri effetti come, ad esempio, l'occlusione ambientale (a meno di non ricorrere agli effetti propri della Global Illumination, come vedremo).



L'illuminazione ambientale nella nostra scena (in questo caso, Skybox con un'intensità particolarmente elevata).

Si tratta di un tipo di illuminazione utile per giochi o applicazioni con grafica stilizzata, in stile cartoon, dove ad esempio si fa uso di materiali

unicolor o di Texture che contengono già effetti di illuminazione (punti più chiari, ombre, eccetera), o in contesti dove non si vuole far proiettare ombre agli oggetti; come detto nel primo capitolo di questa guida, infine, può essere utile in fase di “Blocking in” della scena, ovvero quando ancora si stanno disponendo gli oggetti e si iniziano a provare colori e Textures, osservando il tutto con un'illuminazione uniforme, prima di aggiungere le fonti di luce vere e proprie.

Parametri di base: Source, Intensity, GI

La luce ambientale consente quindi di rendere leggermente visibili gli oggetti, illuminandoli in maniera uniforme ed eventualmente con una tinta di colore, anche in assenza di Skybox (illuminazione dallo sfondo dell'universo virtuale), materiali emissivi o fonti di luce tradizionali.



I parametri di base dell'illuminazione ambientale nella scheda Lighting – Scene.

Le impostazioni di tale tipo di illuminazione globale si trovano nella sezione Scene della finestra Lighting; in particolare, è possibile illuminare la scena con una fonte di luce (e, quindi, di colore) scelta, nel selettore Environment Lighting Source, tra:

- un colore uniforme, con Color e il selettore del colore che apparirà;
- un gradiente, da definire specificando i tre colori nei selettori Ground (terreno), Horizon (orizzonte) e Sky (cielo), per far realizzare l'interpolazione a Unity;
- Skybox, che può essere quindi anche una Texture immagine, eventualmente in HDR.

La scelta di un metodo piuttosto che di un altro dipende più che altro da esigenze di “look” del progetto da realizzare; ovviamente, è possibile disattivare del tutto tale illuminazione: è sufficiente scegliere Color come Source e impostare il colore a nero puro.

Il parametro Intensity Multiplier, disponibile per la modalità Skybox, indica l'intensità dell'illuminazione prodotta dallo sfondo come fonte ambientale nella scena, in un range che va da 0.0 a 1.0, dove le superfici appariranno molto luminose.

Tramite il selettore Ambient Mode è possibile specificare se l'illuminazione ambientale dovrà partecipare alla simulazione di Global Illumination e, quindi, produrre raggi luminosi di rimbalzo e la relativa illuminazione; ciò può avvenire in tempo reale (Realtime) o in maniera statica, precalcolata (Baked); questa opzione rende meno piatta l'illuminazione prodotta da Ambient e, in ogni caso, è necessario selezionare sia la casella Realtime Global Illumination che Baked Global Illumination, più in basso nella scheda Lighting – Scene, per poter attivare questa modalità.

In pratica: esempio di illuminazione ambientale nella nostra scena

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Per poter analizzare l'illuminazione ambientale nella scena d'esempio fornita nel pacchetto Assets, per prima cosa è necessario eliminare tutte le fonti di luce eventualmente presenti nella scena.

Fatto ciò, con la telecamera virtuale posizionata all'interno della cucina (in modo da inquadrare il centro della stanza), aprire la sezione Scene nella scheda Lighting, che nei capitoli precedenti abbiamo posizionato accanto all'Inspector.

Se la casella Auto Generate, in fondo alla scheda, è deselezionata, selezionarla o cliccare sul pulsante Generate Lighting, per far effettuare a Unity l'elaborazione dell'illuminazione statica della scena; mantenendo Auto Generate disattivato, sarà necessario cliccare su Generate Lighting dopo ogni modifica di rilievo alle impostazioni degli oggetti o dell'illuminazione, per valutare i risultati delle azioni.

Nella sezione Environment Lighting impostiamo Source su Color, scegliendo tale opzione dal relativo menù a discesa; le altre opzioni, che vedremo a breve, sono Gradient e Skybox.

Scegliamo quindi il nero come colore per Environment Color. Dal momento che non vi sono fonti di luce nella scena, ci troviamo all'interno di una stanza chiusa con la luce ambientale impostata su nero puro, tuttavia la vista nella finestra Game (l'anteprima della scena inquadrata dalla telecamera) non ci apparirà completamente nera...

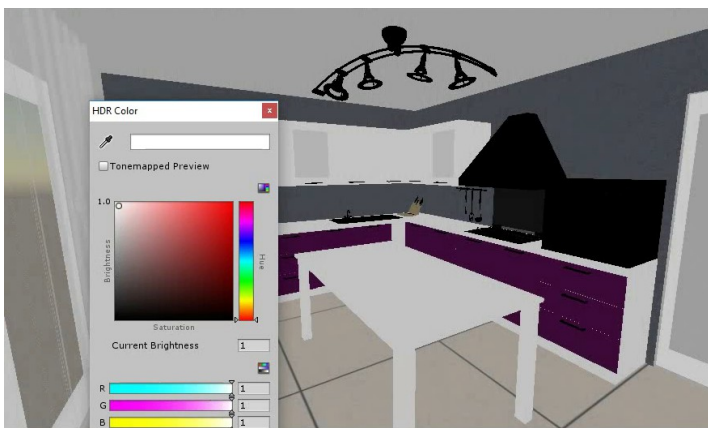


Anche con l'illuminazione ambientale "disattivata" (intensità nulla, colore nero puro), la scena non ci apparirà completamente nera...

Ciò è dovuto alla presenza dei riflessi speculari ambientali: si tratta di riflessioni luminose calcolate, al pari dell'illuminazione ambientale, sulle superfici di tutti gli oggetti; di default, a tutte le superfici della scena vengono forniti i riflessi dell'ambiente esterno, con un'intensità definita dal parametro Environment Reflection – Intensity Multiplier.

Portiamo quindi il valore di questo parametro a 0.0 per eliminare i riflessi ambientali nella nostra scena, che diventerà così completamente nera.

A questo punto, cambiando il colore di Ambient Color tramite il relativo selettore sarà possibile fornire una luminosità di base (espressa come un valore di grigio) o anche una tinta di colore a tutti gli elementi della scena: la luce ambientale è, come detto, un'illuminazione generale di base presente per tutti gli elementi in maniera uniforme.

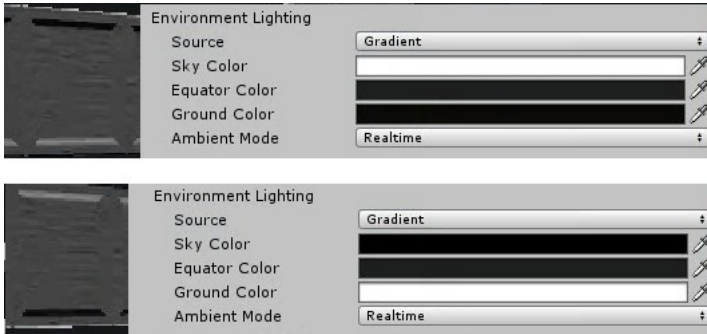


Illuminazione ambientale con Color; gli oggetti metallici appaiono neri perché i riflessi ambientali sono stati portati a 0.0.

Da notare che gli oggetti in metallo, come il frigorifero o il forno, rimarranno neri, per cui per colorare tali superfici con la sola illuminazione ambientale dovremo aumentare il valore dei riflessi speculari ambientali, con Environment Reflection – Intensity Multiplier.

Scegliendo ora Gradient nel selettore Source, i selettori del colore diventeranno tre: Sky, Equator e Ground, ad indicare intuitivamente tre diverse tinte di colore (o di grigio) da applicare alle facce degli oggetti a seconda della loro esposizione: Sky per quelle rivolte verso l'alto, Equator per quelle laterali e Ground per quelle rivolte verso il basso. Si ha quindi la possibilità di definire delle “sfumature” - un gradiente di colore, appunto - per l'illuminazione ambientale.

Guardate, ad esempio, come cambia l'illuminazione delle ante dei mobili impostando lo Sky Color a bianco e gli altri a nero, quindi impostando successivamente il Ground Color a bianco e gli altri a nero: a seconda dei casi, saranno le superfici delle ante rivolte verso l'alto o quelle rivolte verso il basso ad essere illuminate.



L'illuminazione Gradiente mette a disposizione tre selettori che definiranno il colore ambientali per le facce superiori, laterali e inferiori degli oggetti.

Questo gradiente non ha niente a che fare con il colore o la Texture dell'universo virtuale della scena, come è possibile constatare semplicemente spostando la telecamera virtuale all'esterno della stanza: i colori di Gradient definiscono solo l'illuminazione uniforme da applicare direttamente ad ogni oggetto della scena, non definiscono proprietà dell'universo o dello sfondo.

Per poter utilizzare i colori e le sfumature dello sfondo dell'ambiente virtuale come sorgente dell'illuminazione ambientale è necessario cambiare Source in Skybox. Il parametro Intensità, alla pari dei colori nella modalità Gradiente, non ha a che fare con lo sfondo dell'universo virtuale, ma si riferisce esclusivamente alla colorazione uniforme, ambientale, da applicare alle varie facce degli oggetti.

È possibile definire un Material, provvisto di Texture immagine, per lo Skybox; vedremo come farlo nella parte pratica del prossimo capitolo, nel quale ci concentreremo appunto sullo Skybox della scena.

CAPITOLO 4

Lo Skybox

Cos'è uno Skybox e come si crea un Material Skybox

Uno Skybox è sostanzialmente un cubo, ovvero uno spazio virtuale di sei lati, utilizzato come gabbia e all'interno del quale è racchiusa tutta la scena; in un certo senso, è il guscio più esterno della scena e ci consente di definire l'aspetto dell'orizzonte e dello sfondo dell'universo virtuale, che può essere così proiettato su tutti gli oggetti presenti.

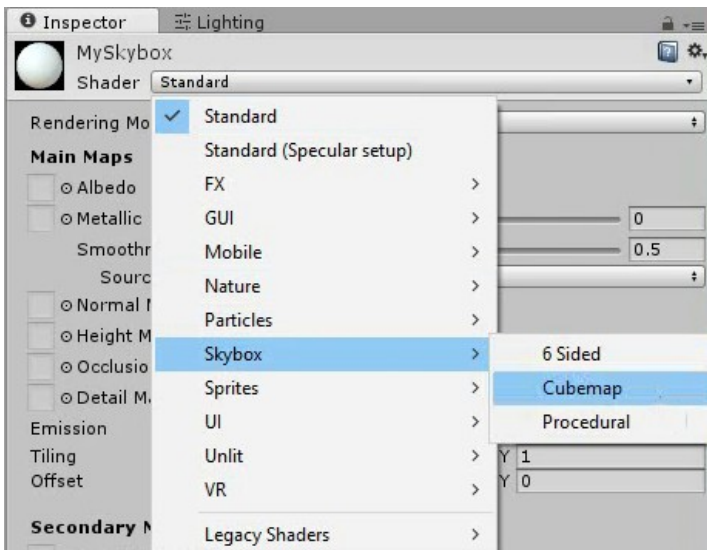
Serve, quindi, a definire uno sfondo per la scena, ma tale sfondo è anche un illuminante: si tratta, in effetti, di una delle tre opzioni utilizzabili per definire l'Environment Source nella scheda Lighting – Scene, utilizzando in particolare il Material di tipo Skybox definito appena sopra, alla voce Environment – Skybox Material.

Unity mette a disposizione un Material Skybox di default; in alternativa, si può definire uno Skybox personalizzato e utilizzarlo per l'illuminazione ambientale della scena con Environment Source impostato proprio su Skybox. In questa fase ci limiteremo a descrivere uno Skybox e la sua creazione; in seguito, parleremo anche dei suoi effetti nell'illuminazione globale in tempo reale in Unity 5.

6-Faces, Cubemap, Procedural; realizzare le immagini (plugin)

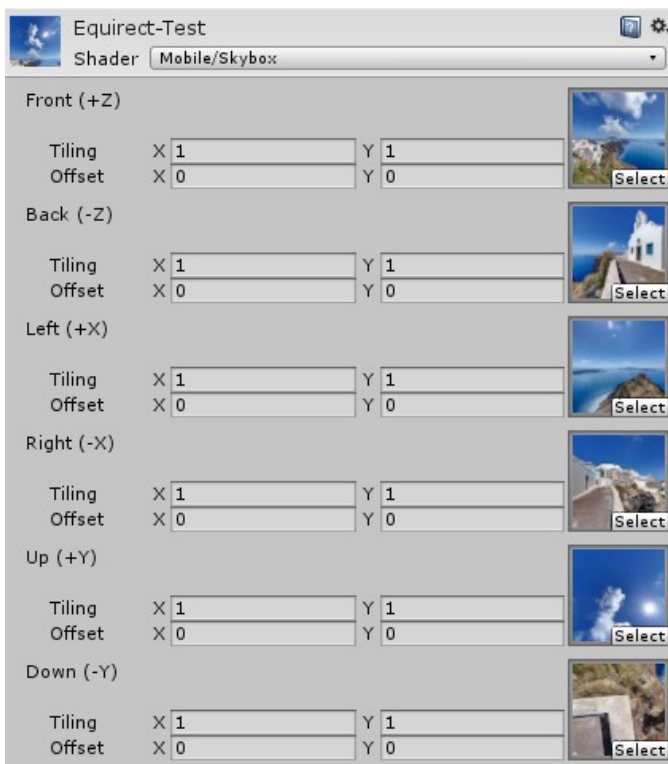
Come detto, lo Skybox è un particolare cubo virtuale, provvisto quindi di 6 lati. Uno Skybox va creato partendo da un Material; in particolare, si crea un nuovo Material per il progetto e si cambia il tipo di ombreggiatore dal selettore Shader (in alto nell'Inspector) passando da Standard a Skybox e scegliendo uno dei tre metodi di definizione di questo tipo di Material.

A seconda del metodo di definizione utilizzato per lo Skybox (tra 6-faces, Cubemap e Procedural), potrebbe essere necessario dover realizzare sei immagini Textures, oppure un'immagine particolare (Cubemap) o dover definire dei parametri, nell'Inspector, per generare uno sfondo virtuale.



Creazione di un materiale Skybox: dopo aver creato un Material nel Project, cambiarne il tipo in Skybox.

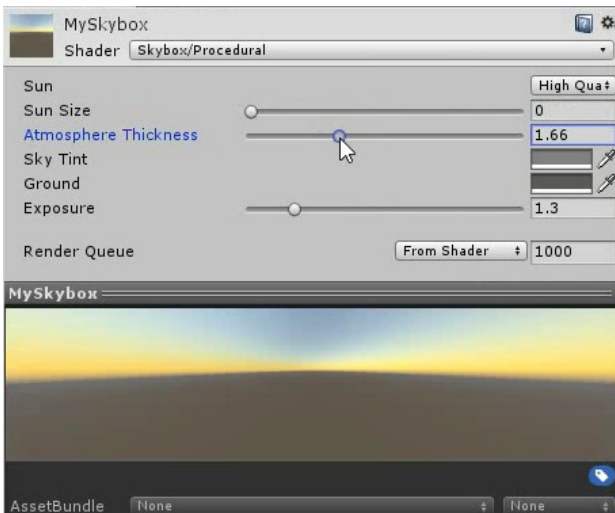
Nel caso della modalità 6-faces serviranno, appunto, sei immagini che corrisponderanno, nello spazio virtuale, al cielo sopra di noi, al terreno sotto di noi e a quattro lati intorno a noi (fronte, dietro, sinistra, destra). A ciascuno di questi lati deve quindi corrispondere una Texture, ovvero un'immagine, caricata nella cartella Assets del progetto e alla quale andrà modificato il Wrap Mode da Repeat a Clamp nell'Inspector. Tale immagine rappresenterà la porzione di sfondo osservabile in quella particolare direzione.



Uno Skybox di tipo 6-faces: le sei immagini rappresentano le viste superiore, laterali e inferiori

Nel caso della modalità Cubemap bisognerà procurarsi, invece, un'unica immagine contenente, in realtà, le 6 immagini dei relativi lati, con tali immagini disposte secondo uno schema ben preciso all'interno dell'unica Texture utilizzata.

In questi primi due casi, le immagini necessarie possono essere realizzate in vari modi; ad esempio, ritagliando fotografie panoramiche, oppure elaborando particolari Textures Mirrorball o Equirectangular, oppure ancora con un programma di grafica 3D (come ad esempio Blender 3D) in grado di realizzare delle Environment Maps di un ambiente virtuale. Nel caso della modalità Procedurale, infine, si potranno variare manualmente o via codice (anche in maniera interattiva a runtime) parametri come l'intensità del Sole, il colore del cielo o quello dello sfondo, in modo da generare uno sfondo virtuale all'interno di Unity. I parametri hanno nomi intuitivi e la Preview in tempo reale, in basso nell'Inspector, consente di valutare le modifiche apportate di volta in volta.



Esempio di Skybox procedurale, con l'anteprima nell'Inspector

Assegnare lo Skybox alla scena corrente

Definito ed impostato uno Shader di tipo Skybox come Asset del progetto come descritto nei paragrafi precedenti, lo si assegna alla scena corrente aprendo la finestra Lighting e trascinando lo Shader nello slot Skybox della scheda Scene; si tratta, appunto, di un elemento che riguarda tutta la scena, non legato ad un particolare oggetto.

Distinzione tra Skybox di telecamera (Clear Flag) e di scena

A questo proposito va detto che lo Skybox impostato in Lighting – Scene non è lo stesso impostato nel campo Clear Flag della Main Camera o, in generale, di una telecamera virtuale.

Nel primo caso, infatti, specifichiamo lo Skybox da utilizzare come illuminante globale nella scena; nel secondo, invece, specifichiamo come renderizzare lo sfondo dell'universo virtuale, ossia come ciascuna telecamera virtuale dovrà colorare lo sfondo, ma togliendo lo Skybox non ne disattiveremo gli effetti di illuminazione (definiti, come detto, in Lighting – Scene).

In pratica: creare uno Skybox (tramite plugin) ed assegnarlo alla scena corrente

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

La definizione di uno Skybox di tipo procedurale è abbastanza semplice: una volta creato un nuovo Material di questo tipo nel Project, lo si potrà personalizzare nel suo Inspector variando parametri dai nomi intuitivi come la dimensione del disco solare, lo “spessore” dell'atmosfera

(Thickness; consente di definire quanto sarà rarefatta e/o torbida l'atmosfera), i colori di base da utilizzare per il cielo e per il terreno (che verranno utilizzati per realizzare il gradiente col quale verrà colorato lo sfondo) e un valore di esposizione, nel senso fotografico del termine.

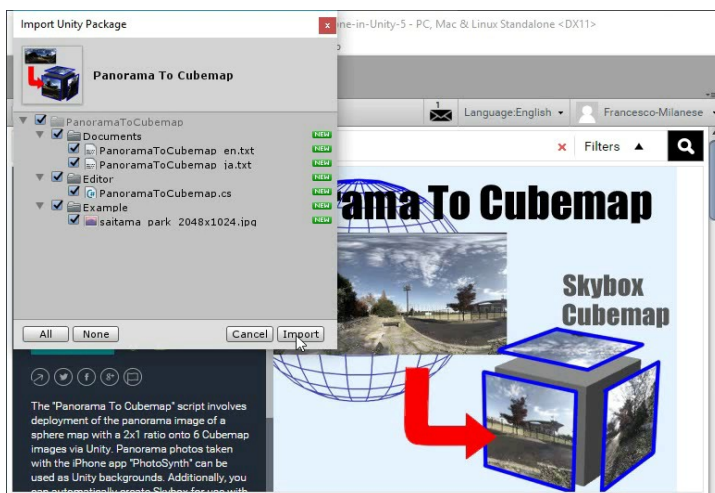
Nella sezione Preview dell'Inspector di uno Skybox procedurale è presente un'anteprima in tempo reale dell'ambiente realizzato modificando i vari parametri, per cui la definizione di un ambiente di base risulta davvero semplice.

I formati accettati da Unity per definire uno Skybox tramite immagini, ovvero 6-faces e Cubemap, non sono propriamente di uso comune; al contrario, le immagini panoramiche salvate in modalità Equirectangular (a 360° in un'unica immagine orizzontale) sono abbastanza diffuse e reperibili gratuitamente, anche in formato HDR, online.

Per questo motivo, Yutaka Yoshisaka ha realizzato un plug-in per Unity chiamato “Panorama to Cubemap”, disponibile gratuitamente nell'Asset Store del programma, che consente appunto di generare una Cubemap o un set di 6 Textures da utilizzare nello Skybox di Unity, sia per lo sfondo della scena che per l'illuminazione ambientale.

Per poter scaricare il plug-in è necessario connettersi all'Asset Store di Unity; per prima cosa, quindi, apriamo una finestra Asset Store nel programma dal menù Window, quindi cerchiamo “Panorama to Cubemap” nella casella di ricerca in alto nella scheda Asset Store.

Il plug-in che cerchiamo dovrebbe comparire in cima ai risultati della ricerca, quindi selezioniamolo e clicchiamo su Import nella relativa scheda; per procedere con il download e l'installazione sarà necessario effettuare il log-in all'Asset Store con il proprio account Unity, anche se il plug-in è completamente gratuito.



Importazione del pacchetto Panorama to Cubemap.

Terminata l'importazione e l'installazione del plug-in, potremo richiamarlo dal menù Window.

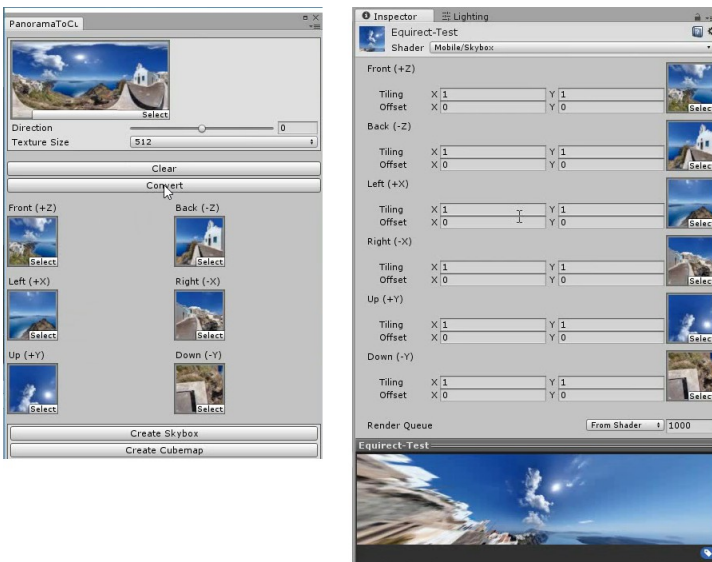
Nella finestra che apparirà a video dovremo impostare, ovviamente, l'immagine sorgente Equirectangular, che andrà quindi importata nel progetto come un qualsiasi Asset e trascinata poi nella finestra nel plug-in, nella sezione Texture2D in alto (per importare e convertire altre mappe, in seguito, cliccare su “Clear”, nella finestra).

Le immagini Equirectangular da convertire devono avere un rapporto 2:1 tra larghezza e altezza.

Cliccando su Convert faremo creare al plug-in le sei immagini che ci servono e queste appariranno nelle relative caselle all'interno della finestra del plug-in.

Cliccando su Create Skybox, poi, il plug-in genererà automaticamente un nuovo Material Skybox, che posizionerà nella cartella output_materials, all'interno di Project, per cui potremo trascinare direttamente quello Skybox nella relativa casella nella sezione Environment Lighting della scheda Lighting – Scene.

Per apprezzare le modifiche nella scena sarà necessario cliccare su Generate Lighting (qualora la casella Auto Generate fosse disattivata).



Cliccare su Convert per generare le sei immagini a partire da quella Equirectangular; quindi su Create Skybox o su Create Cubemap; successivamente, si dovrà impostare lo Skybox così creato come Skybox di scena, nella finestra Lighting - Scene

Il plug-in ci consente anche di generare, cliccando su Create Cubemap, un'immagine Cubemap, all'interno della quale le sei viste vengono posizionate secondo uno schema ben preciso.

Per il momento, riguardo lo Skybox ci fermiamo qui: abbiamo visto come crearne uno e come configurarlo come illuminante per la scena virtuale, distinguendolo tra l'altro dallo sfondo visto dalle telecamere; in seguito, faremo qualche considerazione sugli effetti di illuminazione globale prodotti da tale componente.

CAPITOLO 5

Fonti di luce standard: Directional

Parametri di base (comuni anche ad altre luci): colore, intensità, Bounce Intensity

Le Directional Lights sono fonti di luce standard di Unity caratterizzate, principalmente, da una direzione. La posizione del Game Object nella scena non è importante, perché i raggi luminosi arriveranno “da infinito” lungo la direzione indicata mediante i valori di Rotation. Non è presente, inoltre, l'attenuazione sulla distanza.

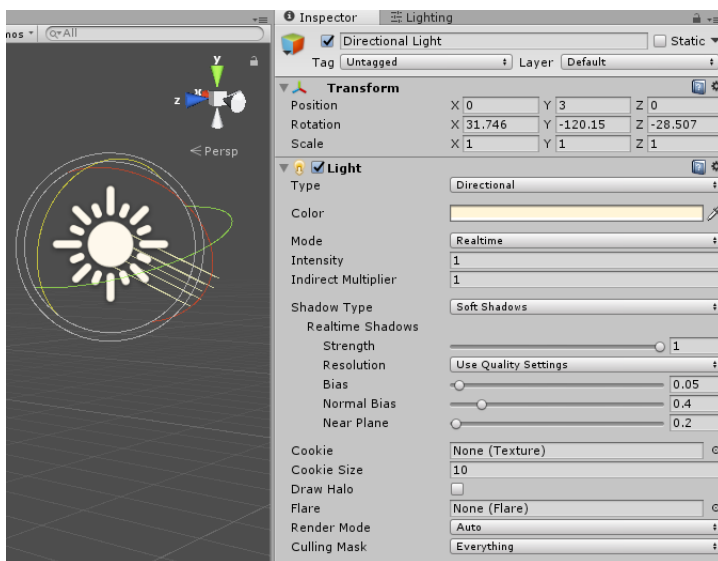
Per tutti questi motivi, Directional si presta bene a simulare la luce solare; in effetti, nel prossimo capitolo parleremo proprio del suo utilizzo per Ambient Light.

Per aggiungere una fonte di luce Directional Light nel progetto o direttamente nella scena corrente è sufficiente fare click destro rispettivamente nella finestra Project o in quella Hierarchy, scegliere Create e, quindi, Light – Directional Light.

Come detto, la posizione non è importante, mentre per l'orientamento possiamo agire sui valori di Rotation XYZ presenti nell'Inspector dell'oggetto; nella finestra Scene l'icona della fonte di luce viene rappresentata con delle linee, proprio per indicare la direzione dei raggi paralleli, mentre nella finestra Game possiamo valutare la variazione dell'illuminazione mediante l'ombreggiatura in tempo reale sugli oggetti.

Nell'Inspector troviamo anche molti parametri comuni ad altre fonti di luce, quali ad esempio il colore, l'intensità luminosa e l'intensità dei

rimbalzi (bounce), ovvero dei raggi luminosi indiretti, prodotti dai rimbalzi dei raggi primari sulle superfici presenti nella scena; quest'ultima, in particolare, è una caratteristica di Global Illumination, che verrà implementata a patto che l'oggetto sia statico o che nella scena vi siano dei particolari oggetti detti Light Probes, come vedremo in seguito.



Una Directional Light (selezionata e con il manipolatore di rotazione attivato) inserita nella scena e i suoi parametri nell'Inspector:

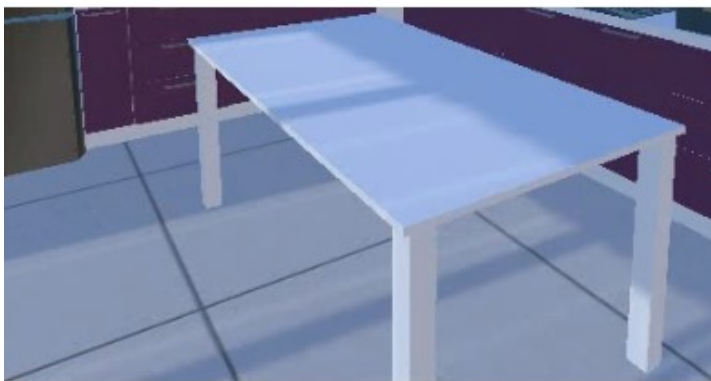
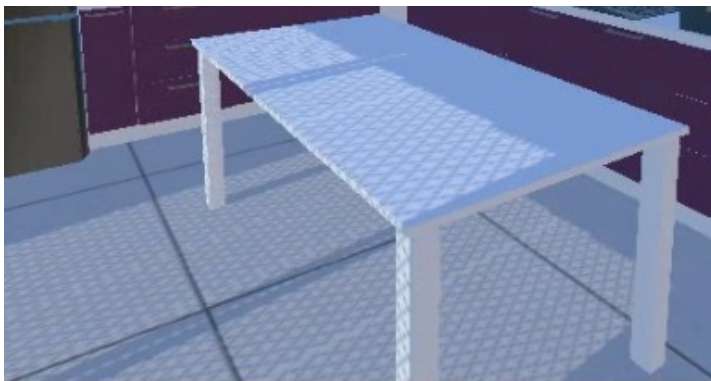
Ombre Hard e Soft; Bias e Normal Bias

Le Directional Light e le altre fonti di luce standard supportano in maniera nativa la proiezione di ombre; tale effetto può essere comunque disattivato o attivato in modalità hard o soft, ovvero con bordi più o meno sfumati, mediante l'apposito selettore nell'Inspector.

In particolare, le ombre hard avranno dei bordi netti: meno costose dal

punto di vista computazionale, ma poco indicate in certi contesti (teoricamente, solo una fonte di luce assolutamente puntiforme dovrebbe proiettare ombre nette e in assenza di atmosfera), mentre le ombre Soft avranno dei bordi sfumati. La qualità delle ombre prodotte va definita, intuitivamente, mediante il parametro Resolution.

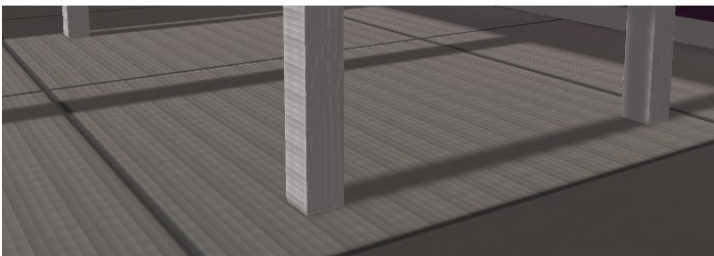
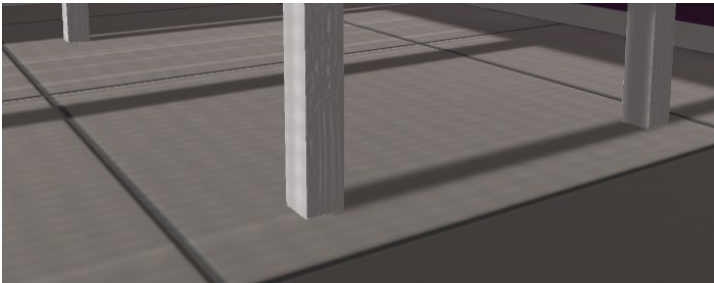
Una differenza notevole tra le ombre Hard e Soft prodotte da una stessa fonte di luce è visibile nella nostra scena d'esempio se si fanno entrare i raggi luminosi dal lato ove sono presenti le tende: con la modalità Hard Shadows, verrà proiettata distintamente la trama del tessuto delle tende sugli oggetti presenti nella stanza, mentre con la modalità Soft queste ombre saranno decisamente più sfumate.



Differenza tra ombre Hard Shadows (in alto) e Soft Shadows (in basso).

A volte, su superfici parzialmente ocluse possono verificarsi degli strani problemi di ombreggiatura, come la formazione di strisce o bande. In molti casi, la loro formazione è dovuta ad un problema tecnico o di calcolo dell'algoritmo che genera le ombre confrontando, internamente, due o più mappe particolari.

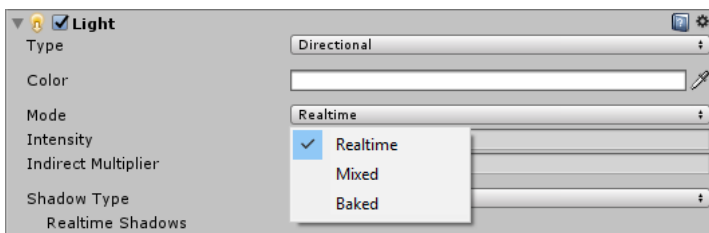
Per risolvere (o quantomeno ridurre) questo problema si può far ricorso ai parametri Bias e Normal Bias, ma bisogna far attenzione a non esagerare perché, in realtà, non si tratta di vere e proprie soluzioni tecniche al problema, quanto di rimedi pratici; in particolare, esagerando con i valori le ombre potrebbero apparire fuori posizione o più sottili di quanto non dovrebbero essere, per cui si consiglia di utilizzare valori molto bassi giusto per evitare la formazione delle bande.



In alto: ombre Soft Shadows e valori Bias e Normal Bias di default; in basso: con i due parametri Bias portati a 0.0; in questo caso, i "quadretti" visibili derivano dalle ombre del tessuto, proiettate sulle superfici all'interno della stanza.

Baking (cenni introduttivi)

Nell'Inspector della fonte di luce è presente anche il selettore Baking, che ci consente di scegliere tra Realtime, Baked e Mixed. Questo selettore serve a definire, quindi, se l'illuminazione dovrà essere precalcolata (Baked) o calcolata in tempo reale.



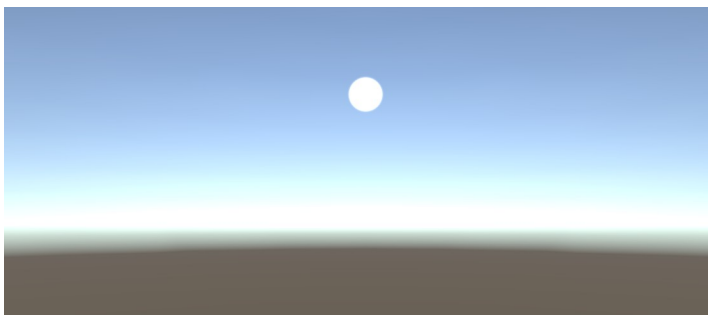
Il selettore Mode, nell'Inspector della fonte di luce.

La modalità Mixed utilizzerà il precalcolo (Baking), ma sarà comunque attiva durante l'esecuzione (Real-time) per illuminare – in maniera diretta, non con luce di rimbalzo – gli oggetti non statici (personaggi o comunque oggetti animati o in movimento) utilizzando le Light Probes.

Parleremo di tutti questi aspetti relativi all'illuminazione indiretta nella seconda parte di questa guida; per il momento, analizziamo solo l'illuminazione diretta prodotta da questa e da altre fonti di luce standard.

Directional come Sun in Ambient Light; orari della giornata; il Sole di una Texture Skybox

Se nella scena vi è almeno una fonte di luce Directional Light, allora essa (o la più intensa delle Directional Lights, qualora ve ne dovesse essere più di una) verrà considerata come il “Sole” dell'universo virtuale e avrà una sua rappresentazione visiva nella scena durante l'esecuzione del gioco, sotto forma di un cerchio bianco.



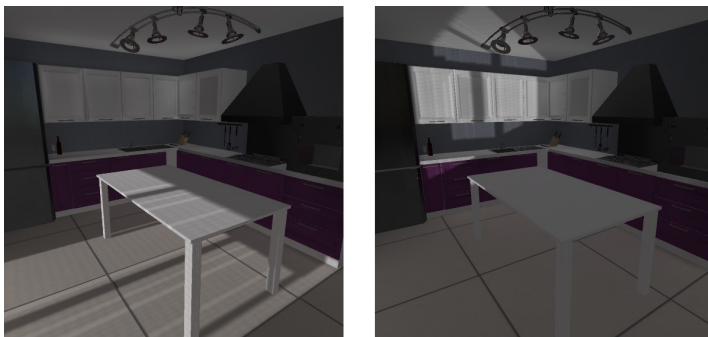
L'icona del Sole, introdotta automaticamente da Unity nella vista Game utilizzando la fonte di luce Directional più intensa presente.

Se vi è una Directional Light Sun, cambiandone l'orientamento si influenzerà anche l'illuminazione dell'intera scena, in quanto l'angolo della Directional Light corrisponderà all'angolo di incidenza dei raggi solari e, quindi, ad un orario della giornata.

Se la posizione dell'oggetto Directional Light non ha quindi alcuna importanza (in quanto i raggi arrivano “da infinito” e non hanno attenuazione), i due parametri importanti sono:

- l'orientamento rispetto all'angolo Y globale della scena, ovvero quello verticale, perché determina il lato ove si trova il Sole;
- l'orientamento rispetto all'angolo Y locale della fonte di luce, perché descrive l'altezza del Sole sull'orizzonte, determinando anche l'orario della giornata e la variazione di illuminazione tra alba, mezzogiorno, tramonto, mezzanotte.

Se si intende utilizzare una fonte di luce direzionale come Sole nell'illuminazione della scena, si dovrebbe rimuovere il Sole reale nella Texture immagine eventualmente utilizzata come Skybox, perché una Directional Light impostata come Sun avrà una sua rappresentazione visiva nella scena e le due immagini (quella della Texture e quella della Directional Light) e i relativi effetti di illuminazione - in particolare, con l'utilizzo di Texture immagine per lo Skybox - potrebbero non combaciare, con tutte le conseguenze del caso.



Per variare la direzione dei raggi solari (e, quindi, l'orientamento della scena e l'orario della giornata), agire sui parametri Rotation X e Y della fonte di luce Directional Light.

In pratica: utilizzare una Directional Light (e Sun) nella scena

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Inserire una Directional Light in una scena è semplicissimo: è sufficiente aggiungerla con Create – Light – Directional Light in Hierarchy (o in Project, se si desidera creare un Prefab da istanziare nelle scene del progetto) ed impostarne i valori di Rotation X e Y per regolare, rispettivamente, l'altezza della fonte di luce (e, quindi, l'orario della giornata) e l'orientamento dei raggi luminosi nella scena, cosa che in effetti ci consente di definire anche i punti cardinali della scena.

In questo momento, gli oggetti della nostra scena sono tutti dinamici e la fonte di luce ha la modalità impostata su Real-Time, per cui un click su “Generate Lighting” nella scheda Lighting – Scene non avrà alcun effetto, dal momento che non ci sono i presupposti per creare l'illuminazione indiretta in maniera statica; per il momento, comunque, analizziamo solo l'illuminazione diretta introdotta da questa fonte di luce nell'ambiente.

Per prima cosa, impostiamo le ombre Shadows su Hard, quindi orientiamo i raggi in modo da farli entrare dal lato ove sono presenti le tende semitrasparenti: sugli oggetti illuminati dai raggi verrà proiettata la trama del tessuto. Cambiando le ombre in Soft Shadows otterremo delle ombre decisamente più sfumate. Possiamo cambiare la qualità di tale effetto agendo sul parametro Resolution.

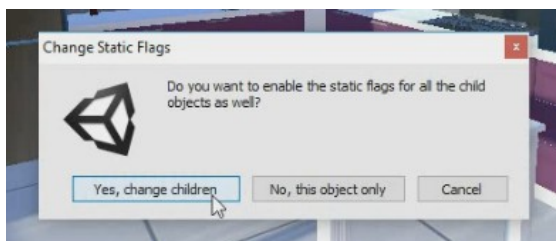


La nostra scena d'esempio con una Directional Light con ombre Hard Shadows attivate; non vi è illuminazione indiretta, ma sono presenti gli effetti di illuminazione e riflessione ambientale.

Analizziamo quindi la variazione dell'illuminazione all'interno dell'ambiente al variare dei valori di Rotation X e Y per la fonte di luce direzionale. Altri parametri di base da impostare riguardano il colore e l'intensità luminosa.

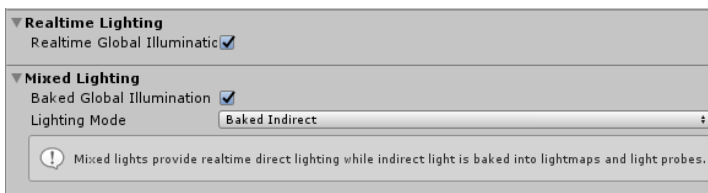
Anche se è ancora presto per parlare dell'illuminazione globale, vediamo un assaggio degli effetti di illuminazione indiretta introdotti con Unity 5. Selezioniamo la fonte di luce Directional Light e cambiamone la modalità in Mixed, nell'Inspector; apparirà il parametro Indirect Multiplier, che indica appunto quanto accentuare l'illuminazione di rimbalzo (indiretta). Lasciamo Indirect Multiplier a 1 e aumentiamo un po', invece, l'intensità della fonte di luce.

Selezioniamo ora la radice della gerarchia di oggetti nella scheda Hierarchy, ovvero l'oggetto KITCHEN, quindi – nell'Inspector – clicchiamo su Static, in alto a destra, e confermiamo la volontà di voler estendere questa caratteristica a tutti gli elementi figli (ovvero: a tutti gli oggetti presenti nella scena).



Dopo aver cliccato su Static nell'Inspector dell'oggetto KITCHEN (la radice di tutta la gerarchia di oggetti della scena), Unity ci chiederà di confermare l'operazione per tutti gli oggetti figli.

Nella finestra Lighting – Scene, adesso, impostiamo Realtime come Ambient Mode (con le caselle Realtime Global Illumination e Baked Global Illumination entrambe selezionate, in basso), poi portiamo Environment Reflections – Intensity Multiplier a 1.0 in modo da generare i riflessi Skybox per le superfici metalliche, quindi cambiamo la modalità Mixed Lighting in Baked Indirect. In questo modo, le fonti di luce contrassegnate come Mixed (come la nostra Directional Light) produrranno un'illuminazione diretta in tempo reale, mentre l'illuminazione indiretta verrà precalcolata e memorizzata per gli oggetti statici (come lo sono tutti gli oggetti della nostra scena).



Nella scheda Lighting - Scene, attivare i due metodi di Global Illumination e cambiare la modalità di Baked in Baked Indirect.

Clicchiamo quindi su **Generate Lighting**, in basso nella scheda; al termine dell'elaborazione, passiamo in modalità **Play** e, da una vista **Game**, osserviamo il risultato ottenuto e la variazione dell'illuminazione diretta in tempo reale, ottenuta variando i valori di **Rotation X** e **Y** nell'Inspector della fonte di luce direzionale.



Il risultato dell'illuminazione indiretta; sono presenti anche i riflessi ambientali, evidenti in particolare sulle superfici metalliche.

Al termine delle nostre prove, torniamo alla sola illuminazione diretta per analizzare le altre fonti di luce standard, nei prossimi capitoli; in particolare, usciamo dalla modalità **Play**, selezioniamo l'oggetto **KITCHEN** e deseleggiamo la casella **Static** nell'Inspector, confermando l'operazione per tutti gli elementi della gerarchia, quindi apriamo il menù accanto al pulsante **Generate Lighting**, in **Lighting – Scene**, scegliendo “**Clear Baked Data**”, per eliminare appunto tutte le informazioni di illuminazione indiretta generate precedentemente.

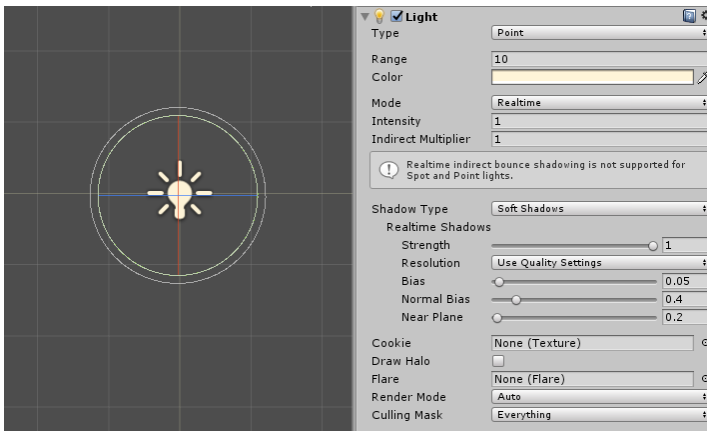
CAPITOLO 6

Fonti di luce standard: Point Light

Parametri: Range, intensità, colore e ombre

Le Point Light sono fonti di luce standard, per cui possiamo crearne una scegliendo la relativa voce dal menù Game Object – Light, dopodiché potremo posizzarla e orientarla nella scena in maniera interattiva o modificando i valori della sezione Transform, nell'Inspector.

Possiamo anche trasformare una fonte di luce tradizionale esistente in Point Light tramite il selettore Type nell'Inspector e questa considerazione vale anche per le altre fonti di luce, ovvero possiamo cambiare il tipo di oggetto selezionato in Directional, Area, Spot o Point.

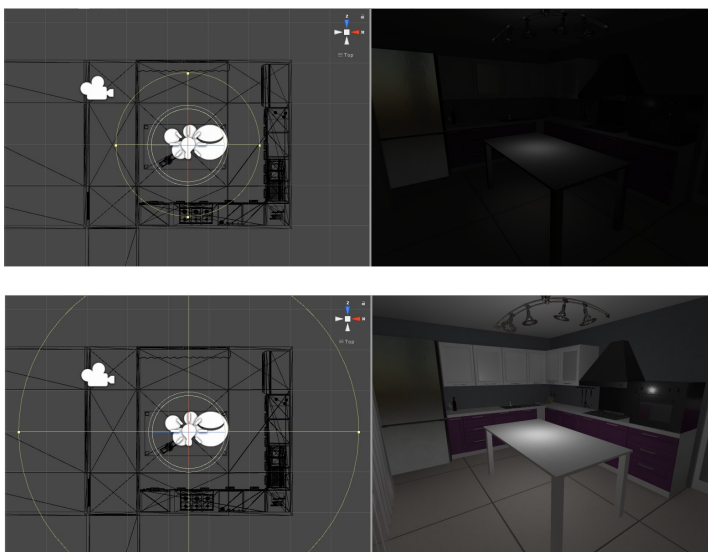


Una fonte di luce Point Light e i suoi parametri nell'Inspector.

Una Point Light è una fonte di luce omnidirezionale, come una lampadina a bulbo, per cui la sua posizione nella scena è importante, in quanto rappresenta l'origine della sorgente luminosa, mentre non lo è la direzione, perché i raggi vanno in tutte le direzioni, differenziandosi così dalla Directional Light sotto entrambi gli aspetti. È indicata, quindi, per lampadine, candele, falò, caminetti, eccetera.

Un parametro che la caratterizza è Range: si tratta della distanza, in unità della scena di Unity, superata la quale non si hanno più effetti di illuminazione da parte della Point Light.

Ciò significa che se un oggetto ci appare come non illuminato, molto probabilmente non servirà a nulla aumentare il valore dell'intensità luminosa, ma piuttosto bisognerà controllare il valore del parametro Range, perché l'oggetto potrebbe trovarsi semplicemente al di fuori del raggio d'azione della fonte luminosa. Il Range è indicato, nella vista Scene, da un paio di cerchi centrati sulla Point Light. All'interno del volume individuato dalla sfera di raggio Range, l'intensità luminosa si attenua fino ad arrivare a 0.



Variazione dell'illuminazione al variare del parametro Range.

Altri parametri sono simili a quelli visti nel capitolo precedente con le Directional Lights: la modalità può essere Baked, Realtime o Mixed.

Il colore della fonte di luce va specificato, intuitivamente, nel campo Color, mentre il selettore delle ombre (Shadows) ci consente di impostare la realizzazione di ombre dure (Hard Shadows, dai bordi netti) o morbide (Soft Shadows, con bordi sfumati).

In pratica: utilizzare una Point Light nella scena; Light Leaks

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

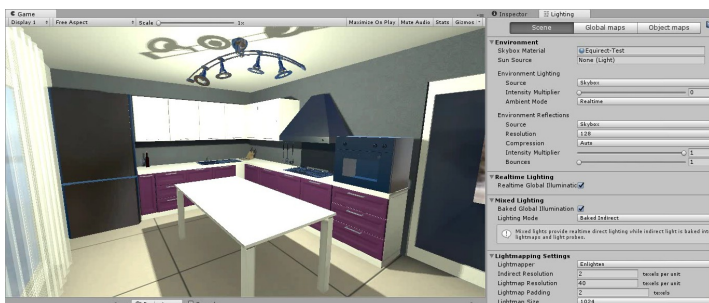
L'inserimento e il posizionamento di una fonte di luce Point Light non sono molto diversi da quelli, visti nel capitolo precedente, effettuati con le Directional Lights, con la differenza che in questo caso dovremo tenere conto della posizione della luce omnidirezionale all'interno della scena e non del suo orientamento.



Point Light con ombre Soft.

Possiamo quindi spostare la luce in tempo reale nella scena, cambiare il tipo di ombre proiettate ed altri parametri di base.

Proviamo quindi l'illuminazione indiretta prodotta da una Point Light di base inserita nella scena, cambiando il tipo della fonte di luce in Mixed, impostando tutti le geometrie in Static ed avviando il primo precalcolo dell'illuminazione indiretta con Generate Lighting, in basso nella scheda Lighting – Scene.



Point Light con Indirect Lighting.

A questo punto possiamo utilizzare la scena d'esempio, distribuita con la versione Assets di questa guida, per analizzare il problema dei Light Leaks, ovvero quello della luce che “fuoriesce” da o su superfici chiuse.

Posizioniamo la fonte di luce Point Light nell'angolo della stanza opposto a quello in cui si trova la telecamera virtuale; inoltre, spostiamo la fonte di luce all'esterno della stanza, al di là della parete, ed aumentiamone di molto il valore dell'intensità luminosa, prima di avviare un nuovo precalcolo con Generate Lighting.

Con l'illuminazione dell'ambiente virtuale (Environment Lighting) ridotta a zero e la fonte di luce Point Light, in teoria non dovremmo vedere nulla all'interno della scena, perché la Point Light si trova all'esterno dell'ambiente... ma in alcuni casi sono visibili alcune strisce di luce che sembrano appunto fuoriuscire da superfici chiuse, in particolare lì dove si hanno facce perpendicolari, come ad esempio tra le pareti ed il soffitto.



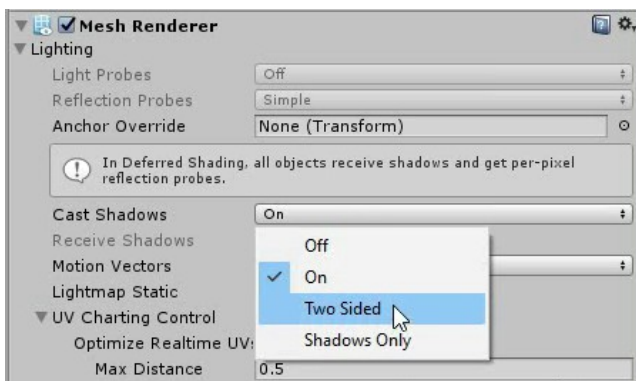
Un esempio di Light Leaks, qui dovute ad un valore di Bias molto alto per la fonte di luce.

Per risolvere questo problema è possibile mettere in atto un paio di accorgimenti; anche se questi non sempre risolvono il problema, spesso si rivelano utili:

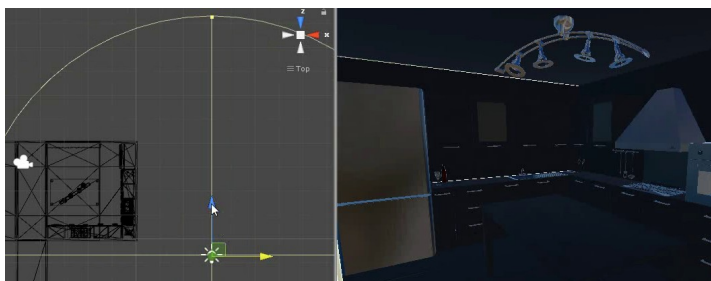
- attivare la proiezione delle ombre per la fonte di luce, se disattivata (l'effetto è infatti accentuato se le ombre sono disattivate);
- portare a 0.0 il valore di Normal Bias per la fonte di luce, ridurre al minimo quello di Bias, entrambi nella sezione Shadows dell'Inspector della fonte di luce;
- per gli oggetti di scena che danno problemi (ovvero, che presentano “perdite di luce”) impostare, nella sezione Mesh Renderer dell'Inspector, la modalità Two Sided per la proiezione delle ombre;
- potete anche agire sul parametro Range della fonte di luce, se questa si trova in un altro ambiente ed è possibile “confinarla” senza problemi nel suo spazio; in effetti, una fonte di luce posizionata all'esterno dell'ambiente A, e che deve illuminare un altro ambiente B, probabilmente può avere un Range d'azione limitato al proprio ambiente B, per cui si può agire sul parametro Range della fonte di luce per evitare che questa “sfori” nell'ambiente A, che vogliamo preservare.

Prima di passare al prossimo capitolo, disabilitiamo l'illuminazione indiretta nella scena, ma questa volta procediamo in questo modo: cambiamo la modalità della fonte di luce in Real-Time, quindi

disabilitiamo sia Realtime Global Illumination che Baked Global Illumination nella scheda Lighting – Scene e, in fondo in questa scheda, clicchiamo su Generate Lighting. A differenza di quanto fatto al termine del capitolo precedente, quindi, stiamo lasciando gli oggetti di scena su Static e stiamo disattivando le tecniche di illuminazione globale.



Il menù Cast Shadows, nel Mesh Renderer di un oggetto, consente di impostare la modalità Two Sided per la proiezione delle ombre, utile per ridurre le Light Leaks.



Ridurre il Range delle fonti di luce che si trovano in altri ambienti per ridurre le Light Leaks.

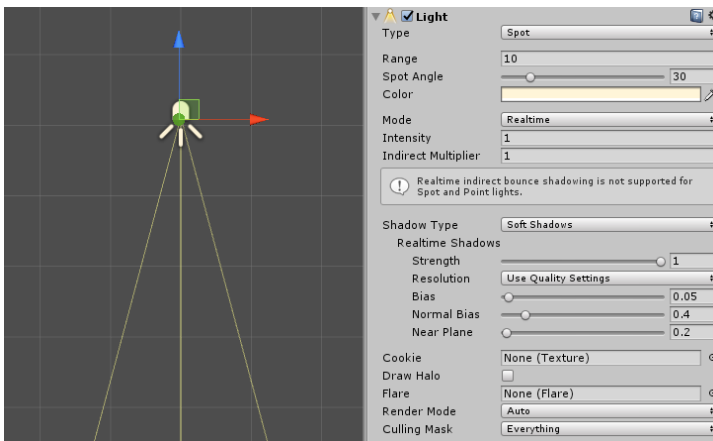
CAPITOLO 7

Fonti di luce standard: Spot Light

Parametri di una Spot Light

Le Spot Light sono fonti di luce simili a faretti, ovvero hanno un punto di origine come le Point Lights e una direzione come le Directional Lights ma, a differenza delle altre, hanno anche un cono di apertura, che determina la porzione di spazio illuminata intorno al vettore della direzione.

L'ampiezza del cono di apertura viene definita utilizzando il parametro Spot Angle, nell'Inspector della fonte di luce; quando il valore è pari a 179 gradi si realizza, sostanzialmente, una Area Lamp (ma Real-Time).



Una Spot Light e i suoi parametri di base nell'Inspector:

Anche in questo caso abbiamo infatti un'attenuazione luminosa sulla distanza ed è presente il parametro Range, per cui l'intensità dell'illuminazione (parametro Intensity) è pari a 0 se si supera la distanza specificata (in questo caso, lungo la direzione indicata dall'orientamento della Spot Light nella scena).

Come per le Point e altre fonti di luce, possiamo specificare colore, ombre e altri parametri di base; valgono quindi le stesse considerazioni fatte nei capitoli precedenti

In pratica: utilizzare una Spot Light nella scena

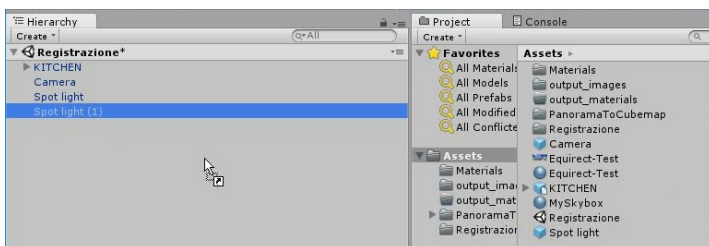
Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Nella scena d'esempio fornita col pacchetto Assets possiamo posizionare almeno una fonte di luce Spot Light in prossimità di uno dei faretto del lampadario; in caso di più Spot Lights, per dare a tutte le fonti le stesse caratteristiche di base possiamo crearne una sola nel Project, anziché nella Hierarchy, in modo da creare un Prefab, che poi istancieremo quattro volte nella scena, trascinando quattro volte il Prefab da Project ad Hierarchy, quindi posizionando ed orientando singolarmente le quattro fonti di luce così create.

Procediamo in questo modo:

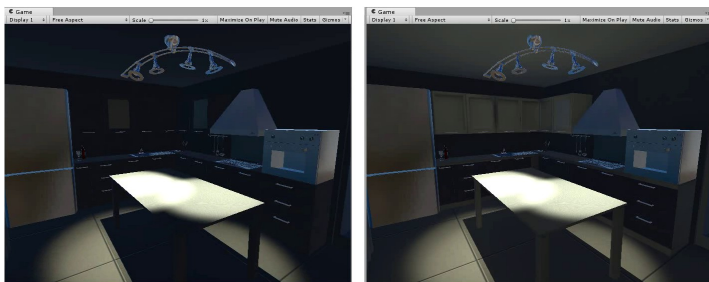
1. aggiungiamo una Spot Light nella scena con click destro in Hierarchy – Create – Light – Spot Light;
2. trasciniamo questa Spot Light da Hierarchy a Project;
3. cancelliamo la Spot Light originale da Hierarchy;
4. personalizziamo il Prefab della Spot Light nel Project, in particolare cambiando la modalità in Mixed (in modo da far calcolare anche l'illuminazione indiretta sugli oggetti statici, durante il Baking), attivando le Soft Shadows, impostando poi 0.0 per Bias e Normal Bias ed impostando un Range ed

- un'apertura adeguati alla scena che vogliamo illuminare;
5. istanziamo il Prefab nella scena, trascinandolo da Project a Hierarchy tutte le volte che vogliamo (o via codice, mediante script, a runtime), quindi impostiamone posizione e orientamento nella scena.



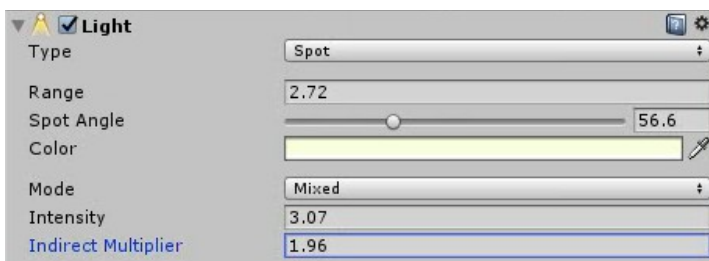
Istanziare il Prefab della Spot Light, dopo averne impostato i parametri, trascinandolo dal Project alla finestra Hierarchy.

Per analizzare la differenza tra l'illuminazione della stanza con e senza illuminazione indiretta, ottenuta mediante Baking delle luci e oggetti statici, possiamo osservare la scena con le fonti di luce appena posizionate che illuminano, di default, in tempo reale, quindi procedere attivando le due tecniche di illuminazione globale in Lighting – Scene, disattivate al termine del capitolo precedente, cliccando poi su Generate Lighting per far effettuare il calcolo dell'illuminazione indiretta a Unity.



Illuminazione prodotta da Spot Light: solo diretta (a sinistra) e con illuminazione indiretta (a destra).

Qualora la scena, illuminata solo dalle Spot Lights, dovesse risultare troppo scura, si potrà agire sul parametro Indirect Multiplier, nell'Inspector delle fonti di luce (o, nel nostro caso, del Prefab, in modo da applicare le modifiche a tutte le fonti di luce istanziate), per poi effettuare un nuovo precalcolo (con Generate Lighting) con le impostazioni aggiornate.



Il parametro Indirect Multiplier nell'Inspector di una fonte di luce in modalità Realtime o Mixed.

Anche se l'utilizzo di una Spot Light può sembrare una buona idea per i farette presenti nel lampadario, va detto che questa soluzione non illuminerà proprio le lampadine, perché poste dietro al cono di ciascuna Spot Light! Ci sono vari modi per risolvere questo problema e uno di essi consiste nell'adottare un Material Emissive per i bulbi delle lampadine: ce ne occuperemo in uno dei prossimi capitoli, trattando proprio questo tipo di materiale, anche con variazione di luminosità in tempo reale via script.

CAPITOLO 8

Introduzione all'illuminazione globale statica e real-time in Unity 5

Illuminazione indiretta e Global Illumination

Eccoci finalmente alla principale novità introdotta con Unity 5 riguardante l'illuminazione della scena: la Global Illumination in real-time, ovvero l'implementazione, oltre che dell'illuminazione diretta in real-time (presente anche nelle versioni precedenti di Unity), anche di quella indiretta (di rimbalzo) e dei relativi effetti in tempo reale nelle scene; in particolare, gli effetti di illuminazione indiretta introducono un maggior grado di realismo perché consentono di illuminare in maniera più convincente le parti non direttamente esposte alle fonti di luce, ma che ricevono comunque dei fotoni di rimbalzo, introducendo nel frattempo effetti particolari come il “color bleeding”, un fenomeno che si ha quando due oggetti di due colori molto diversi sono molto vicini e sull'uno si proietta una tinta derivante dal colore dell'altro, proprio per via dell'illuminazione indiretta.

In Unity, l'illuminazione diretta (in arrivo da fonti di luce standard) è in real-time, ma finora non si aveva la possibilità - se non tramite qualche truccetto - di implementare gli effetti di illuminazione indiretta in tempo reale. Vediamo quindi come tale problema è stato gestito in Unity.

Illuminazione indiretta statica: Baked e Precomputed

La Global Illumination implementata con un algoritmo di ray tracing,

ovvero proiettando dei raggi per i vari fotoni sia diretti che di rimbalzo (ed eventualmente con un discreto numero di rimbalzi e di raggi secondari proiettati ad ogni rimbalzo) è, chiaramente, computazionalmente onerosa, per cui è necessario ricorrere a delle approssimazioni per realizzarla in applicazioni in tempo reale.

Unity presenta due soluzioni a questo problema, per gli oggetti statici e per quelli dinamici: Baked GI e Precomputer Realtime GI.

Per gli elementi della scena che non devono essere modificati, non devono cambiare le loro proprietà e non devono essere animati – ovvero, per tutti gli oggetti statici – è possibile precalcolare gli effetti di illuminazione indiretta e applicarli in maniera definitiva a tali elementi con un processo noto come Baked GI (letteralmente, “illuminazione globale cucinata”, nel senso di già pronta, già realizzata). Questi oggetti potranno ancora ricevere ombre e altri effetti di illuminazione diretta a runtime, ovvero durante l'esecuzione; tali effetti verranno quindi miscelati con l'ombreggiatura definita in maniera statica in fase di Baking.

Il risultato del Baking, per ogni oggetto, è una Texture detta Lightmap, contenente le informazioni di illuminazione indiretta, che verrà poi miscelata con l'illuminazione diretta da Unity, in tempo reale, durante l'esecuzione del gioco. Realizzando quindi una Lightmap, statica, non ci sarà bisogno di implementare gli effetti di illuminazione indiretta in tempo reale: tali effetti verranno calcolati solo una volta, durante il baking e, poiché gli oggetti non si muoveranno, restituiranno una rappresentazione credibile, apportando un notevole risparmio in termini di risorse e di calcoli da effettuare in real time.

Illuminazione indiretta dinamica: Light Probes, Reflection Probes

Per poter illuminare con illuminazione indiretta anche gli oggetti dinamici, mantenendo comunque accettabile il costo computazione anche su sistemi non di fascia alta, Unity utilizza un approccio di approssimazione chiamato Precomputed Realtime. Questa soluzione prevede, in realtà, il precalcolo di tutti i possibili rimbalzi luminosi

all'interno della scena e memorizza i dati, in una forma “approssimata” (come un modello semplificato della scena) per poi utilizzarli a runtime. Durante l'esecuzione, quindi, tutte queste informazioni vengono miscelate con l'illuminazione Baked e quella diretta per ottenere l'ombreggiatura finale degli oggetti.

È possibile definire degli effetti di illuminazione indiretta in tempo reale sugli oggetti dinamici mediante le Light Probes (lett.: “sonde della luce”): sfere virtuali nello spazio che, durante la fase di baking della scena, raccolgono informazioni sull'illuminazione indiretta prodotta da fonti di luce statiche e, come vedremo in seguito e negli esempi pratici, la applicano agli oggetti dinamici che si trovano nel loro campo d'azione. Anche se Precomputed è una buona soluzione per la GI in tempo reale, presenta una limitazione notevole quando si parla di illuminazione in arrivo da oggetti dinamici (ad esempio, un oggetto con Material emissivo) o di effetti dinamici (ad esempio, un oggetto dinamico non creerà color bleeding su uno statico – mentre il viceversa è possibile grazie alle Light Probes).

Allo stato attuale di hardware e software è necessario sottostare a questa limitazione se si desidera, come nel caso di Unity, poter implementare degli effetti di Global Illumination in real time anche su sistemi di fascia (medio-)bassa.

Impostazioni delle Lightmaps (Baked) e Lightmap Parameters

È possibile attivare o disattivare la simulazione sia per Baked GI che per Precomputed GI a livello di progetto mediante le apposite sezioni della finestra Lighting – Scene. Questa possibilità è utile perché, sebbene sia possibile utilizzare entrambi gli strumenti contemporaneamente, ciascun metodo richiede una discreta quantità di risorse e di tempo per l'elaborazione (seguendo una pipeline di elaborazione dedicata, per cui con entrambi i metodi si tratterà di seguirle entrambe, in sequenza); per esempio, per distribuire il gioco su dispositivi mobile potrebbe essere utile disattivare Precomputed Realtime GI, che invece può essere

attivato (anche insieme a Baked) nelle distribuzioni per dispositivi desktop o console.

Nella sezione Lightmapping Settings della finestra Lighting – Scene è possibile specificare, al campo Lightmap Parameters, l'insieme di parametri e caratteristiche delle Lightmaps statiche e della loro creazione. Ciò può essere utile, ad esempio, per realizzare dei “presets” di impostazioni da utilizzare in situazioni diverse, come scene in notturna o scene in pieno giorno, in interni / esterni, eccetera.

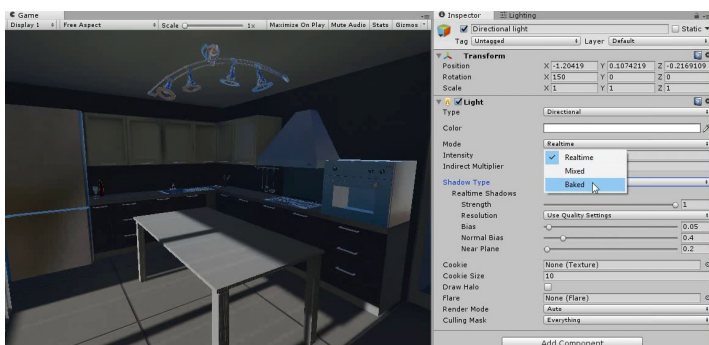
I singoli insiemi di parametri sono degli Assets di scena veri e propri chiamati Lightmap Parameters, che possono essere creati nel Project con click destro – Create – Lightmap Parameters, rinominati a piacere ed impostati scegliendoli dal menù a scomparsa di Lightmap Parameters (in Lightmapping Settings).

In pratica: impostazioni Lightmap Parameters

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

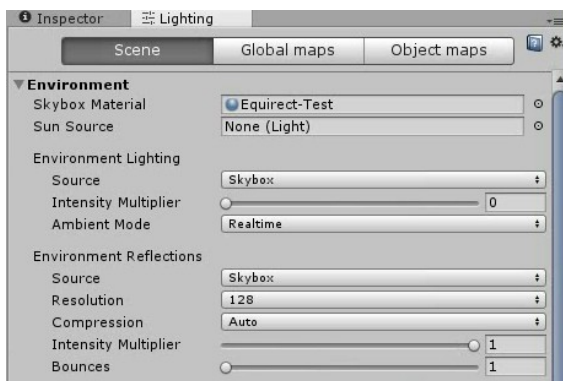
Nei capitoli precedenti abbiamo fatto qualche test di illuminazione indiretta utilizzando le fonti di luce standard e mantenendo i parametri con valori di default; impostiamo ora un sistema di illuminazione indiretta Baked, per i soli elementi statici, esaminando inoltre parametri della scheda Lightmapping Parameters, che contengono alcuni parametri relativi anche al Baking dell'illuminazione diretta della scena.

Iniziamo con una scena contenente i modelli 3D e, per quanto riguarda l'illuminazione, una fonte di luce direzionale, orientata in modo da far entrare i raggi luminosi dall'apertura con le tende (Rot X e Y a 150.0 e 0.0 rispettivamente). Attiviamo le ombre Soft Shadows e cambiamo la modalità della fonte di luce in Baked.



La scena con fonte di luce Baked.

Nella scheda Lighting – Scene portiamo l'Intensity Multiplier dell'illuminazione ambientale a 0.0, in modo da concentrarci sull'illuminazione generata dalla sola fonte di luce direzionale (se non consideriamo i riflessi ambientali, che manteniamo, con Environment Reflection – Intensity Multiplier a 1.0).

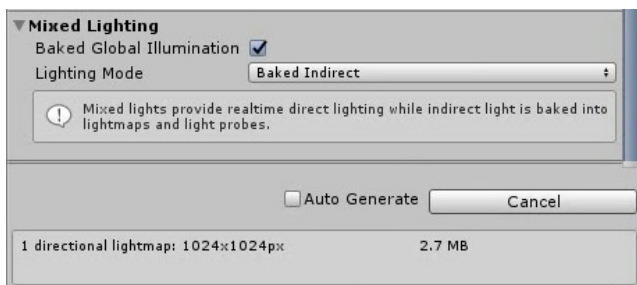


Impostazioni di illuminazione e riflessi ambientali in Lighting - Scene.

Disattiviamo Realtime Lighting: in questa prima fase ci concentreremo sulla sola illuminazione Baked su oggetti statici; a tal proposito, rendiamo statici tutti gli oggetti nella scena (se non lo sono già), come visto nei capitoli precedenti.

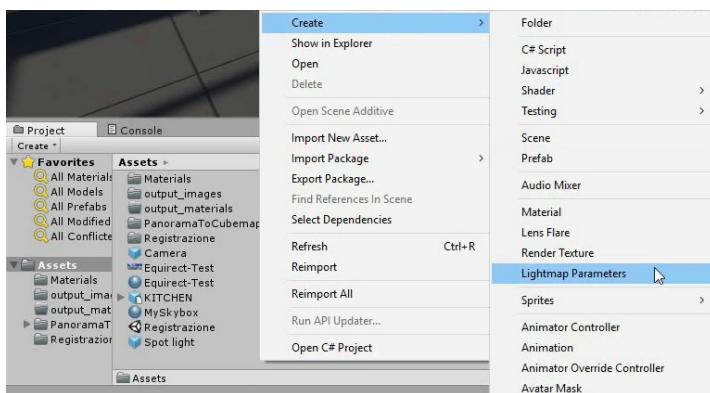
Attiviamo quindi Baked Global Illumination in Mixed Lighting e cambiamo il Lighting Mode in Baked Indirect: in questo modo, come ci suggerisce la casella di testo posta sotto il menù, l'illuminazione indiretta verrà precalcolata (baked) e memorizzata nelle Lightmap degli oggetti statici o, se presenti, nelle Light Probes, per gestire anche gli oggetti dinamici (non presenti, in questo capitolo).

Prima di passare alla creazione dei Lightmap Parameters e alla loro descrizione, attiviamo la casella Auto Generate in fondo alla scheda Lighting, in modo da far effettuare a Unity il primo Baking della scena e tutti i successivi Baking, ogniquale volta modificheremo un parametro relativo all'illuminazione diretta o indiretta, in maniera automatica.



Attiviamo Baked Global Illumination e avviamo il primo Baking con Generate Lighting, in fondo alla scheda.

Creiamo ora un oggetto di tipo Lightmap Parameters nel Project, con click destro in tale finestra e Create – Lightmap Parameters, chiamandolo ad esempio “Kitchen-Low”, ad indicare che questo elemento conterrà le impostazioni per l'illuminazione indiretta di qualità bassa (Low Quality o Low Resolution). Nel menù di Lightmap Parameters, nella scheda Lightmapping Settings di Lighting – Scene, troveremo ora anche questo preset, insieme a quelli nativi di Unity, per cui potremo cambiare velocemente preset per valutare i diversi risultati del calcolo dell'illuminazione indiretta; scegliamo “Kitchen-Low” in questo menù, quindi selezioniamo tale Asset in Project e passiamo alla finestra Inspector per analizzarne i parametri.

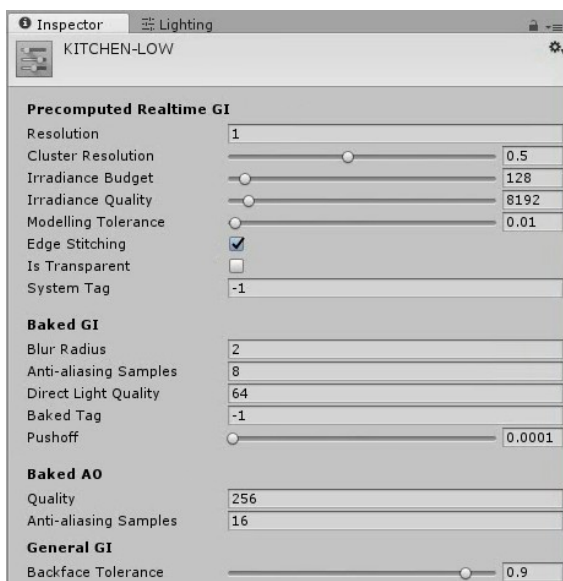


Creare un oggetto Lightmap Parameters.

In questo capitolo ci stiamo concentrando sulla sola illuminazione Baked, per cui esaminiamo solo la sezione Baked GI di questo particolare oggetto:

- **Blur Radius:** è un parametro relativo ad una sfocatura che viene applicata all'illuminazione diretta; in particolare, indica l'ampiezza dell'intorno di texels (i "pixel delle textures", in questo caso della Lightmap) da considerare, per ciascun punto della Lightmap, sui quali effettuare una media dei valori, in modo da sfocare la Texture; un valore troppo elevato può sfocare eccessivamente la mappa, con effetti evidenti soprattutto in prossimità delle ombre; valore di default: 2 (texel);
- **Anti-Aliasing Samples:** l'anti-aliasing è il ben noto filtro anti-scalettatura che, in questo contesto, agisce sulla mappa per smussare bordi e linee; intuitivamente, a valori maggiori si avranno risultati di qualità superiore, ma a costo di un maggior tempo di Baking; valore di default: 8 (texel);
- **Direct Light Quality:** indica il numero di raggi luminosi da utilizzare in un'area, per calcolarne l'illuminazione; come per il parametro precedente, ad un valore maggiore corrisponderanno risultati migliori (soprattutto nel calcolo delle Soft Shadows) ma al prezzo di un maggior tempo di Baking; valore di default:

- 64 (raggi luminosi per area);
- **Baked Tag:** un valore intero che forza Unity ad utilizzare Lightmap diverse per alcuni oggetti, anche se questi hanno parametri uguali a quelli di altri oggetti (NOTA: per ogni oggetto è possibile specificare un Lightmapping Parameter da utilizzare, con tale impostazione che sovrascriverà quella globale di Lighting – Scene, tramite il parametro Lightmap Parameters nella sezione Mesh Renderer – Lightmap Setting, nell'Inspector dell'oggetto); generalmente, per Baked Tag va bene il valore di default, pari a -1;
- **Pushoff:** un valore di spostamento delle geometrie durante la fase di Ray Tracing, utile in presenza di punti neri, artefatti o altri errori nella generazione delle ombre o dell'occlusione ambientale, soprattutto su superfici ampie; valore di default: 0.0001 (unità di misura della distanza nella scena).



La scheda di un oggetto Lightmap Parameters.

CAPITOLO 9

Baked e Precomputed GI

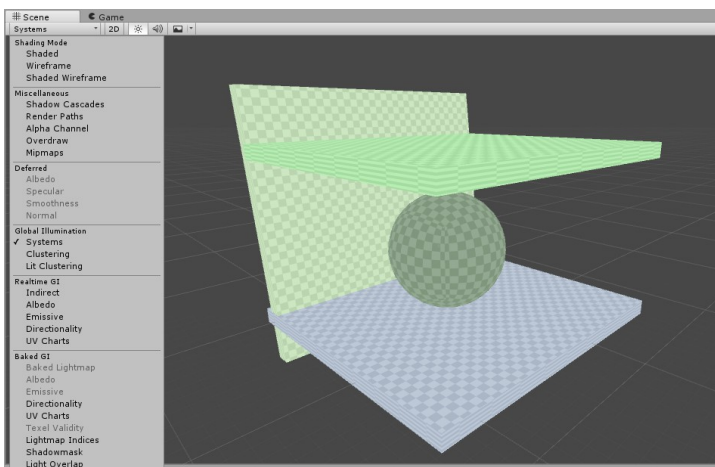
Enlighten, Systems e precalcolo

Enlighten è un motore di Radiosity, ovvero un modulo software che calcola l'illuminazione globale in tempo reale in una scena 3D, presentando un approccio particolare al problema della complessità di tale elaborazione, la quale consiste nel risolvere l'equazione del rendering, ovvero l'insieme di calcoli da effettuare per elaborare l'illuminazione in una scena.

Lo scopo è quello di realizzare gli effetti con vari livelli di resa e accuratezza su vari tipi di piattaforme, riducendo il più possibile alcuni problemi noti dell'Illuminazione Globale (come quello della convergenza dei rendering) realizzati con i metodi utilizzati dal programma.

Un'immagine non convergente presenta infatti del “rumore” e, in caso di animazioni o esecuzioni in real-time (con più immagini renderizzate al secondo) tale rumore può introdurre altri problemi, come sfarfallii o altri effetti spiacevoli. Per poter realizzare la Global Illumination su sistemi che vanno dagli smartphone alle console e ai sistemi desktop di fascia alta, Enlighten suddivide gli spazi in isole (dette *systems*), introduce vincoli e restrizioni in base alle caratteristiche e alla potenza di software e hardware di destinazione ed elabora il problema in due modi: in parte, in maniera precalcolata (baked, statica) e in parte in maniera dinamica (real-time). Ciò è necessario per poter distribuire l'applicazione sul maggior numero possibile di tipi di dispositivi.

In particolare, durante una fase di precalcolo Enlighten suddivide una scena in *systems*, elaborate poi in parallelo, con più *tasks* o passi di elaborazione per ogni isola. Tale suddivisione viene effettuata automaticamente da Enlighten, ma è possibile agire parzialmente su come vengono generati i *systems*. Al termine del precalcolo vengono valutate le relazioni, in termini di illuminazione, tra i vari *systems*. Quanto detto si ricollega al discorso sulle limitazioni di Enlighten in Unity: per poter realizzare la GI in tempo reale, buona parte delle scene va trattata in maniera STATICA e il setup di ambienti e *systems* deve essere noto a priori. Vediamo quindi come Enlighten tratta, invece, la parte dinamica delle scene.



I Systems individuati da Unity in una semplice scena.

Enlighten consente di trattare dinamicamente (in tempo reale durante l'esecuzione del gioco o programma):

- fonti di luce standard;
- illuminazione ambientale;
- materiali emissivi.

In seguito vedremo, con esempi concreti, come usare queste luci in modalità real-time GI.

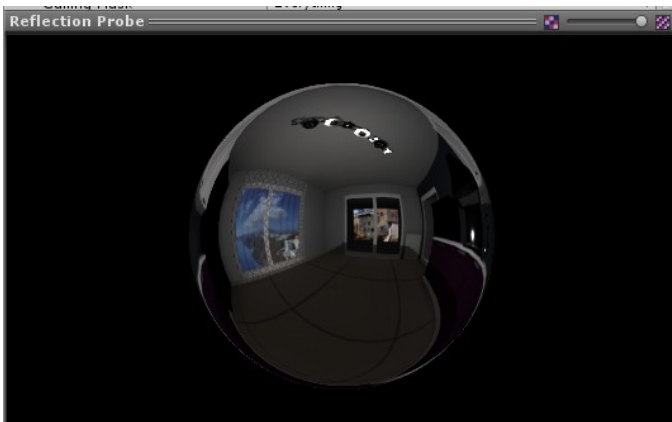
Le fonti di luce appartenenti a questi tre tipi possono essere infatti contrassegnate, con appositi selettori, come Baked o come Realtime.

Se vengono trattate come Baked, allora Enlighten produce delle Lightmaps, come avveniva nelle versioni precedenti di Unity, ovvero delle Textures contenenti informazioni sull'illuminazione e le ombre presenti in maniera statica (precalcolata) per i vari oggetti.

L'integrazione tra illuminazione Baked e illuminazione in Realtime (GI) viene gestita automaticamente da Unity a runtime.

In particolare, nel corso dell'elaborazione della scena, Enlighten fa uso di:

- Light Maps - per l'illuminazione diffusa degli oggetti statici;
- Light Probes - per l'illuminazione diffusa degli oggetti dinamici;
- Cube Maps - per supportare i riflessi Glossy / Specular.



Una Reflection Probe è un oggetto di scena di Unity che "fotografa" l'ambiente circostante e utilizza l'immagine Texture così creata per generare i riflessi speculari sugli oggetti presenti all'interno del suo campo d'azione.

Baked e Precomputed Real-Time; impostazioni in Lighting

All'interno della sezione Lightmapping Settings di Lighting – Scene troviamo ora:

- **Lightmapper:** consente di scegliere il modulo software che si occuperà della generazione delle Lightmaps (Enlighten, nel nostro caso);
- **Indirect Resolution:** un valore, espresso in texel per unità della scena, che indica quanti texel della Lightmap riservare per ogni “metro quadro” (o, meglio, “unità quadrata”) di superficie; il valore è MOLTO più basso di quello utilizzato per la Lightmap Resolution del metodo Baked GI, perché in questo caso memorizziamo solo l'illuminazione indiretta nelle mappe, non l'illuminazione diretta o le ombre; il valore di default è 1 e generalmente va bene per molte scene, ma può diminuire a 0.5 o anche 0.1 per scene ampie e terreni, oppure aumentare a 2-3 per scene piccole, interni (come nel nostro caso) o comunque scene con molti dettagli;
- **Lightmap Resolution:** come per Indirect Resolution, ma in questo caso per tutta l'illuminazione Baked, quindi inclusa anche l'illuminazione diretta e le ombre – e, per questo motivo, i valori qui utilizzati sono generalmente MOLTO maggiori di quelli visti per il parametro precedente (di default è 40, ma può essere anche molto maggiore);
- **Lightmap Padding:** il Padding è uno spazio utilizzato per separare due o più elementi; in questo caso, si riferisce ad uno spazio da inserire tra le superfici scucite nella Lightmap;
- **Lightmap Size:** intuitivamente, la risoluzione in pixel (qui espressa indicando la misura di un solo lato, visto che le Lightmap sono sempre quadrate) della Texture Lightmap generata; a valori maggiori corrisponderanno mappe più definite, ma anche tempi di calcolo più lunghi per il Baking e una maggiore quantità di risorse necessarie a runtime, per utilizzare le mappe;
- **Compress Lightmap:** come suggerisce il nome, consente di

comprimere le Textures Lightmap, riducendo ovviamente lo spazio necessario per memorizzarle, ma in alcuni casi la compressione può dar luogo a difetti e artefatti nella mappa;

- **Ambient Occlusion:** consente di includere l'occlusione ambientale in fase di Baking e, quindi, nelle mappe generate; è utile per scurire parti occluse o dettagli come crepe, fori, tagli ed altro, ma può essere sostituito dall'effetto SSAO (Screen Space Ambient Occlusion) implementato dal pacchetto Post-Processing Stack, che descriveremo nell'ultimo capitolo di questa guida;
- **Final Gather:** metodo che, se attivato, migliora la qualità finale del Baking, al costo ovviamente di un maggior tempo di elaborazione richiesto; attivando tale modalità appariranno dei parametri supplementari che consentiranno di definire il numero di raggi da utilizzare per ogni punto del processo (e, intuitivamente, anche in questo caso qualità e tempo richiesto saranno direttamente proporzionali) e se utilizzare o meno il Denoising, ovvero un filtro supplementare che, come suggerisce il nome, “ripulisce” (“de-noise”, rimuove il rumore, nel senso di artefatti puntiformi nell'immagine finale) le Textures Lightmap;
- **Directional Mode:** se impostato su Directional, farà memorizzare a Unity le informazioni sulla direzione dell'illuminazione ambientale nelle Lightmap generate da Baked o da Precomputed GI; valore di default: Directional (non supportato, però, su tutti i sistemi, per cui in alcuni casi Unity passerà automaticamente a Non-Directional);
- **Indirect Intensity:** un moltiplicatore dell'intensità luminosa dell'illuminazione indiretta, valido sia per Baked GI che per Precomputed Realtime; valore di default: 1; provate a modificarlo anche in modalità Play, dopo aver calcolato l'illuminazione della scena;
- **Albedo Boost:** valore che può accentuare (Boost) l'intensità dei rimbalzi luminosi tra le varie superfici, in particolare incrementando la luminosità della Texture del colore principale degli oggetti (Albedo); il valore di default, 1, è corretto dal punto di vista fisico, mentre a valori maggiori le superfici

tenderanno a diventare bianche, perché sempre più luminose; provate a modificarlo anche in modalità Play, dopo aver calcolato l'illuminazione della scena.

Ci occuperemo dei Lightmap Parameters (ultima voce di questa sezione di Lighting) nei prossimi capitoli.

Il Baking e le Lightmaps

Ormai è chiaro che il Baking dell'illuminazione è un procedimento che consente di pre-calcolare, prima della compilazione e dell'avvio del gioco, gli effetti di illuminazione diretta e indiretta sugli oggetti statici nella scena virtuale, applicando poi tali ombreggiature e illuminazioni agli stessi oggetti sotto forma di Texture (dette Lightmaps), in modo da simulare quegli effetti senza dover, per questo, calcolarli nuovamente ad ogni frame durante l'esecuzione del gioco.

L'illuminazione può quindi essere distinta in diretta (in arrivo dalle fonti di luce standard; calcolata sempre in real-time, con la proiezione di ombre, senza raggi di rimbalzo ma solo sulle superfici direttamente esposte alla luce) e indiretta (illuminazione di rimbalzo, con raggi secondari).

Nelle versioni precedenti di Unity, l'illuminazione indiretta poteva essere implementata solo per gli oggetti statici tramite il Baking; dalla versione 5, come vedremo in seguito, sono stati introdotti dei sistemi di illuminazione globale in tempo reale.

Come detto, il Baking funzionava (e funziona, in effetti) solo per oggetti statici: dal momento che l'illuminazione e le ombre erano pre-calcolate e applicate come Textures, uno spostamento degli oggetti o una loro deformazione avrebbe “svelato il trucco”.

Questo è, in effetti, l'unico accorgimento (o vincolo, se vogliamo) da tenere a mente quando si effettua un Baking: può essere fatto per oggetti statici, che magari potranno essere messi in ombra da altri (in quel caso

tramite gli effetti di illuminazione diretta, che sono sempre in real-time), ma dovranno essere immutabili, non dinamici.

Lightmapping Bake nelle versioni precedenti e in Unity 5

L'illuminazione indiretta andava quindi precalcolata (“Baked”) tramite gli appositi strumenti nella finestra Lightmapping – Bake, dove era possibile specificare, tra le altre cose, il numero di rimbalzi di illuminazione indiretta da calcolare, la loro intensità, il numero di raggi, la risoluzione ed altri parametri relativi alla qualità dell'effetto.

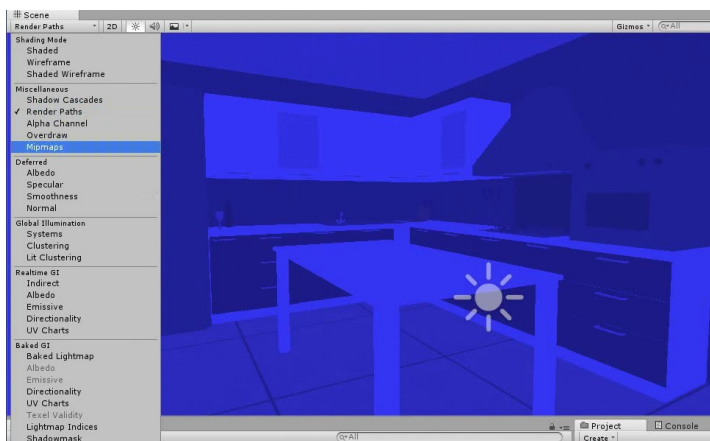
In Unity 5, il Baking di ombre e luci dirette è integrato nel processo di Baking GI, ovvero la serie di operazioni che riguarda anche il precalcolo dell'illuminazione globale (con luce indiretta ed effetti collegati). I relativi comandi si trovano, quindi, nella finestra Lighting – Scene, come abbiamo già avuto modo di vedere.

Degli oggetti vengono quindi realizzate, anche in questa versione, delle Lightmaps da applicare agli stessi sotto forma di Textures di ombreggiatura, che Unity miscelerà automaticamente all'illuminazione diretta, al materiale e alle Textures proprie dell'oggetto per ottenere il risultato finale durante l'esecuzione del gioco.

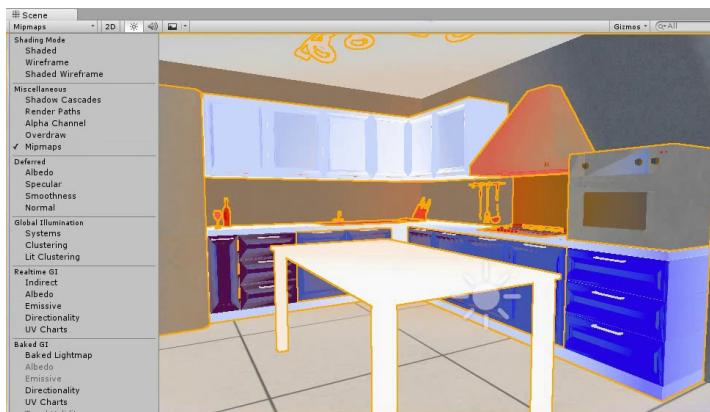
In pratica: visualizzare i Cluster (e non solo) nell'editor

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Nella scheda Scene dell'editor è presente, in alto a sinistra, un menù a discesa che consente di selezionare la modalità di visualizzazione da utilizzare all'interno della finestra.



Modalità di visualizzazione della scena nella finestra Scene.



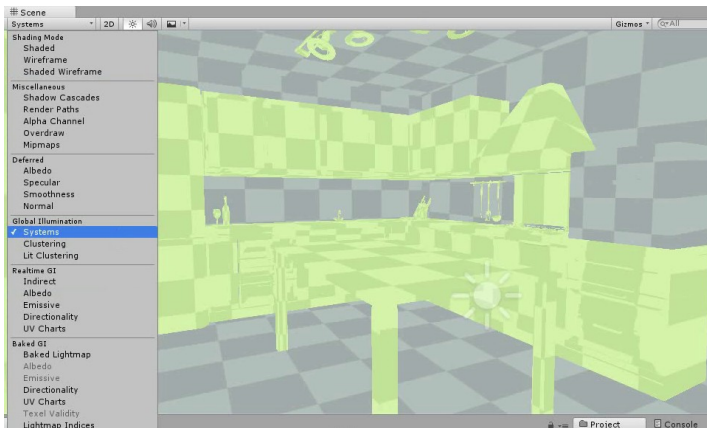
La modalità di visualizzazione Mipmap evidenzierà in rosso gli oggetti con Textures inutilmente grandi (dal punto di vista di osservazione corrente), in blu quelli le cui Textures potrebbero avere risoluzione maggiore.

Utilizzando le voci di tale selettore è possibile ottenere, ad esempio, delle rappresentazioni ombreggiate (Shaded) o struttura (Wireframe) della scena, così come delle mappe rappresentative delle modalità di Rendering Path adottate nella scena, delle dimensioni delle Textures con

MipMaps, eccetera; ad esempio, con Rendering Path il blu indicherà che verrà utilizzato il sistema Deferred, mentre con MipMap, a seconda della distanza alla quale verranno inquadrati gli oggetti, questi verranno evidenziati in rosso qualora dovessero avere Textures inutilmente grandi, in blu in caso di Textures che potrebbero avere risoluzione maggiore.

Le rappresentazioni relative alle informazioni di illuminazione, sia precalcolata che real-time, si trovano nei gruppi Global Illumination, RealTime GI e Baked GI, all'interno del menù.

Per visualizzare i Systems e i Clusters individuati da Unity all'interno della scena, scegliere le relative voci; attenzione, però: i Systems vengono individuati da Unity nella fase di precalcolo, relativa agli oggetti statici, per cui gli elementi devono essere contrassegnati, appunto, come Static.



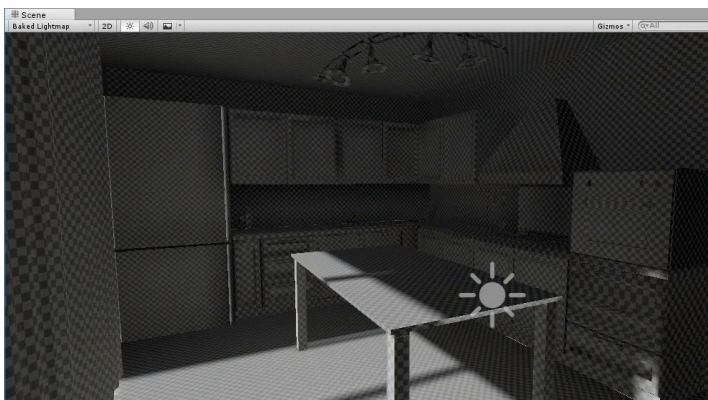
I Systems individuati da Unity nel nostro progetto.

La scena d'esempio fornita nel pacchetto Assets di questo corso contiene invece degli elementi che, all'inizio, sono tutti dinamici: nessuno di essi è contrassegnato come Static, nell'Inspector; in effetti, passando alla modalità di visualizzazione Systems o Clustering con il Package Unity

caricato per la prima volta in un nuovo progetto, non vedremo nulla nella vista Scene.

Selezioniamo quindi l'oggetto genitore della gerarchia, in Hierarchy, ovvero KITCHEN, quindi clicchiamo sulla casella Static nell'Inspector, confermando la volontà di estendere l'impostazione Static a tutti gli elementi figli di KITCHEN nella finestra che apparirà a video.

A seconda che il Baking (il precalcolo) sia attivo o meno per l'intero progetto, potremmo vedere delle modifiche o... nulla; passiamo quindi alla finestra Lighting – Scene, assicuriamoci di avere sia Baked che Realtime Global Illumination attive e selezioniamo, in basso, la casella “Auto Generate”, per far effettuare a Unity il Baking della scena ad ogni modifica che riguarda, appunto, il precalcolo dell'illuminazione.



Visualizzazione Baked Lightmap - DOPO aver effettuato il Baking.

Unity effettuerà quindi il Baking, perché rileverà l'impostazione Static degli oggetti, che appariranno adesso all'interno della vista Scene; in particolare, con la modalità di visualizzazione Systems l'oggetto avrà una texture uniforme, perché in effetti si tratta di una struttura semplice, chiusa, per cui Enlighten lo individuerà appunto come un unico volume, mentre in visualizzazione Clustering vedremo più zone individuate dall'algoritmo nelle varie parti della superficie.

Ogni volta che inseriremo o modificheremo un oggetto come Static, Enlighten eseguirà automaticamente i vari calcoli ed individuerà i Systems e i Clusters; il consiglio che vi do è quello di disattivare l'Auto Generate, per risparmiare tempo quando dovete fare tante modifiche agli elementi statici della scena, premendo invece “Generate Lighting”, che trovate accanto a “Auto Generate”, in basso nella scheda Lighiting – Scene, quando volete aggiornare il Baking della scena.

CAPITOLO 10

La fonte di luce Area Light e il suo baking

Baked (Static)

Le fonti di luce Area rappresentano, come suggerisce il nome, delle superfici emissive.

Si tratta cioè di fonti luminose nei quali i raggi partono da una superficie e non da un unico punto, come le Spot e le Point, ma nemmeno da una superficie infinita o “da infinito”, come le Directional; inoltre, sono orientate verso una direzione.

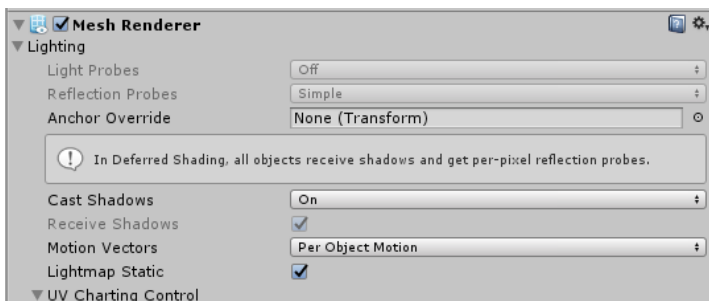
Anche se non vi è un parametro Range, come per le Spot o Point Lights, anche le Area Lamps producono un'illuminazione che presenta attenuazione sulla distanza.

Nelle versioni precedenti di Unity, l'illuminazione in arrivo da una Area Light doveva essere precalcolata (Baked). Per le varie mesh da illuminare andava quindi effettuato il Bake sulle Textures Light Maps. Con la versione 5 di Unity è possibile applicare gli effetti di illuminazione di un'Area Lamp anche agli oggetti dinamici; prima, però, vediamo come realizzare l'illuminazione Baked con le nuove impostazioni.

Creata, posizionata e impostata l'Area Lamp specificando i parametri di intensità luminosa, colore ed altre caratteristiche di base nell'Inspector, il lavoro riguarda in realtà gli oggetti che vogliamo sottoporre ad illuminazione statica.

Bisogna quindi selezionare tali geometrie ed impostarle a Static, o

quantomeno selezionare la casella Lightmap Static nella sezione Mesh Renderer – Lighting nell'Inspector di ciascun oggetto.



L'impostazione Lightmap Static nella sezione Mesh Renderer di un oggetto di scena.

Con Auto Generate attivo in Lighting – Scene, Unity effettuerà il Baking dell'illuminazione sulla geometria e, al termine, la stessa ci apparirà illuminata e ombreggiata.

In pratica: Baking di una fonte di luce Area Light nella scena d'esempio

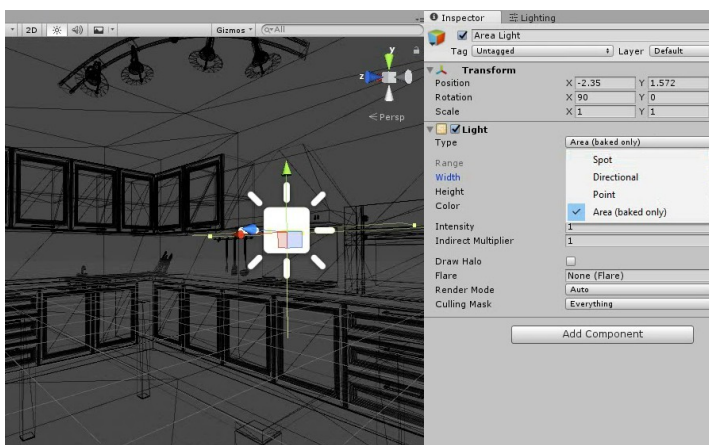
Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Le Area Light prevedono solo la modalità Baked, per cui non possiamo valutarne gli effetti in tempo reale in fase di editing ma dobbiamo effettuare almeno un Bake e procedere, successivamente, con qualche modifica all'intensità della fonte di luce o dei rimbalzi (illuminazione indiretta), nell'Inspector.

Nei capitoli precedenti abbiamo impostato a Static tutti gli oggetti della scena e abbiamo selezionato la modalità Baked Global Illumination in Lighting – Scene, disattivando inoltre l'illuminazione ambientale ma mantenendo i riflessi ambientali con intensità 1.0; manteniamo queste impostazioni, eliminiamo eventuali fonti di luce nella scena ed

inseriamo una Area Lamp nella stessa, posizionandola ed orientandola ad esempio lì dove dovrebbe trovarsi la luce della cappa, sopra i fornelli.

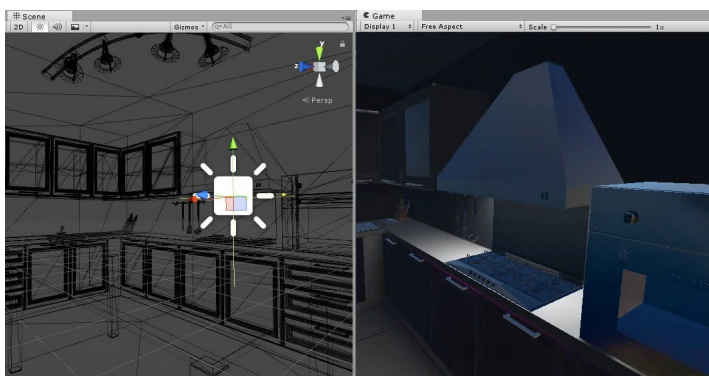
Nell'Inspector della nuova fonte di luce notiamo subito che questa è disponibile solo in modalità Baked, per cui impostiamo colore, intensità luminosa ed intensità dell'illuminazione indiretta (Indirect Multiplier) prima di avviare un Baking dell'illuminazione della scena.



La fonte di luce Area, che può essere solo Baked, inserita nella scena.

Disponendo solo della modalità Baked, le fonti di luce Area non proiettano illuminazione diretta, né quindi ombre di oggetti dinamici su altri oggetti della scena: inserendo adesso un qualsiasi oggetto tra la fonte di luce Area e una superficie illuminata (dopo il Baking) non vedremo né effetti di illuminazione sul nuovo oggetto né ombre sulle altre superfici.

Le fonti di luce Area sono quindi prettamente indicate per fornire illuminazioni diffuse o per progetti di architettura e visualizzazione statica in generale, dove non dovranno essere inseriti oggetti a real-time o in movimento.



L'illuminazione prodotta dopo il Baking della fonte Area Light.

CAPITOLO 11

Illuminazione indiretta con Precomputed Realtime

Precomputed Realtime GI in Unity 5

Il Baking statico, trattato nei capitoli precedenti, genera delle Textures Lightmaps per gli oggetti statici della scena, per cui l'illuminazione viene fissata in fase di realizzazione del progetto e non può essere cambiata in tempo reale.

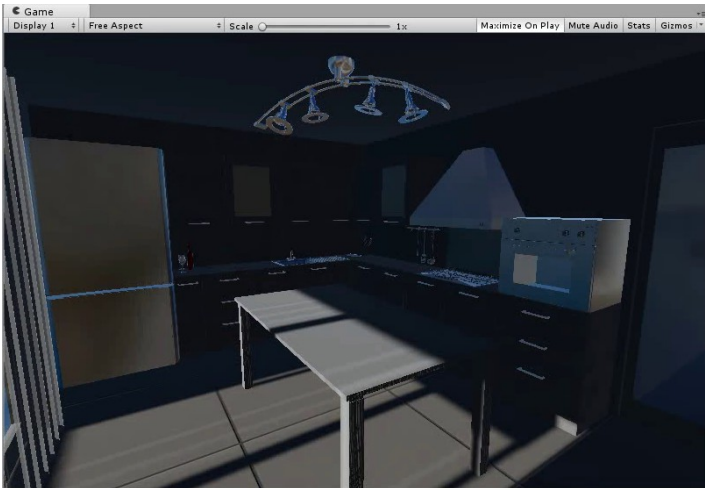
La modalità Precomputed Realtime di Unity 5 memorizza invece alcune informazioni sull'illuminazione della scena in particolari Assets; il calcolo e l'elaborazione di queste informazioni avviene, come nel caso del Baking, prima dell'esecuzione, tuttavia le informazioni di Precomputed Realtime sono strutturate in modo tale da poter essere modificate durante l'esecuzione; ecco quindi il significato del nome di questo metodo: Precomputed perché il precalcolo avviene, in effetti, durante la fase di progettazione, ma Realtime perché le informazioni così generate possono essere utilizzate e modificate in tempo reale, durante l'esecuzione del gioco.

I tempi di precalcolo e la qualità delle informazioni sull'illuminazione generata con Precomputed Realtime dipendono, ovviamente, dai parametri impostati per tale metodo di illuminazione; passiamo quindi direttamente ad un esempio pratico per mostrare i parametri più importanti di Precomputed Realtime e approfittarne per qualche osservazione sulla loro importanza e sui criteri da seguire per scegliere dei valori appropriati.

In pratica: illuminare una scena con Precomputed Real-Time

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Utilizzando la scena d'esempio, fornita nel pacchetto Assets di questa guida, inseriamo solo una fonte di luce direzionale con valori Rotation XY 150.0 0.0, in modo da far entrare la luce dall'apertura con le tende, cambiamo la modalità in Realtime e impostiamo le Soft Shadows.

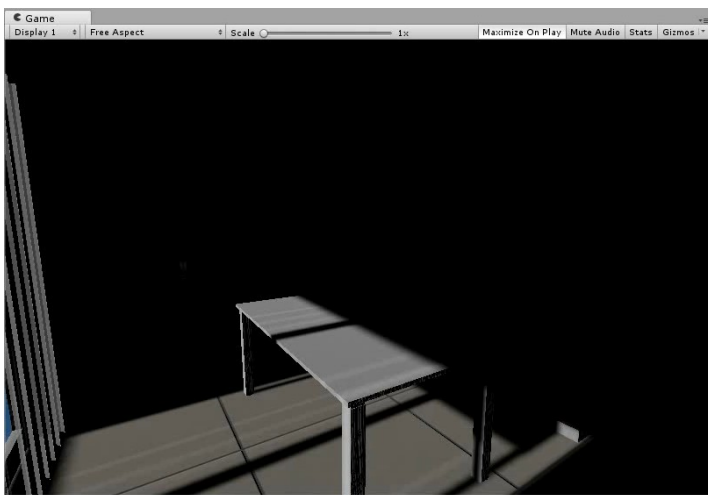


La scena di partenza.

Gli oggetti della scena dovranno essere Statici. Vedremo come fornire illuminazione indiretta anche agli oggetti dinamici tra un paio di capitoli, quando tratteremo le Light Probes.

Nella scheda Lighting – Scene eliminiamo sia l'illuminazione che i riflessi ambientali portandoli a 0.0; ciò avrà l'effetto di rendere nere le

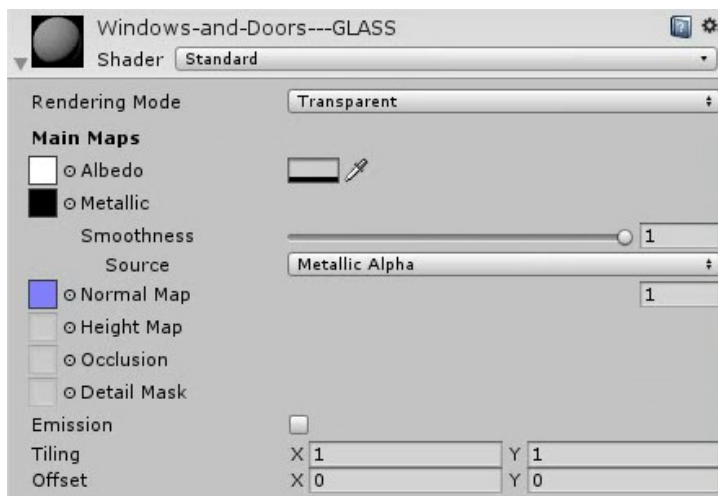
superfici metalliche, ma per il momento vogliamo analizzare la sola illuminazione diretta e indiretta, calcolata da Precomputed Realtime, in arrivo dalla fonte di luce direzionale, senza altre distrazioni.



Dopo aver annullato gli effetti ambientali e disattivato Baked GI.

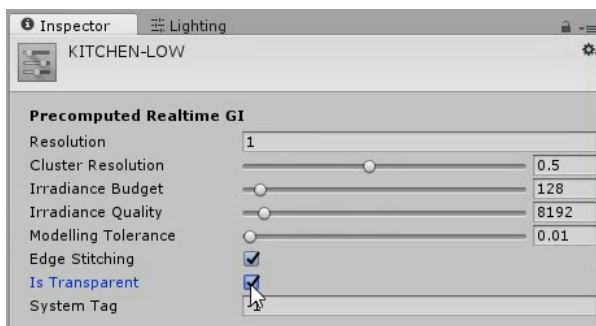
Disattiviamo Baked Global Illumination e attiviamo Realtime Global Illumination, quindi puliamo la Cache dell'illuminazione indiretta con Clear Baked Data (nel menù accanto a Generate Lighting), infine attiviamo Auto Generate in modo da far fare automaticamente a Unity i precalcoli dell'illuminazione ad ogni cambiamento di rilievo.

Anche se Unity effettuerà il Baking per Precomputed, nella vista Game non vedremo alcuna illuminazione indiretta sugli oggetti presenti nella stanza. Il motivo è che gli oggetti Windows-and-Doors presenti vengono visti come solidi dal motore di calcolo dell'illuminazione indiretta, per cui – anche se l'illuminazione diretta riesce a passare dal materiale semitrasparente delle vetrate – i raggi dell'illuminazione indiretta vengono bloccati da tali oggetti e non illuminano l'interno della stanza. Variare i valori di Indirect Intensity e di Albedo Boost (che descriveremo a breve) non risolve il problema.



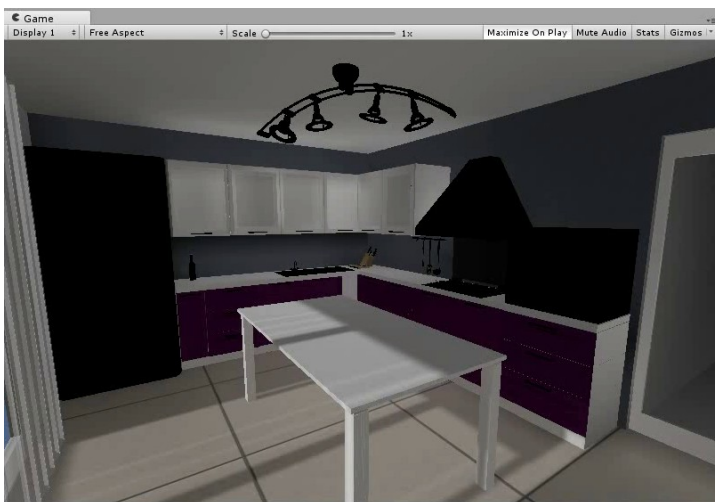
Il sottomateriale Glass di Windows-and-Doors è semitrasparente, ma viene visto come opaco da Precomputed Realtime.

La soluzione è poco elegante, ma c'è: clicchiamo su Edit accanto al preset selezionato per Lightmap Parameters, in Lighting – Scene, in modo da aprire tali parametri nell'Inspector, quindi selezioniamo la casella “Is Transparent” nella sezione Precomputed Realtime GI; in questo modo, gli oggetti appariranno trasparenti durante la proiezione e il calcolo dell'illuminazione indiretta, con l'effetto che potranno raggiungere gli oggetti posti all'interno della stanza e illuminarli.



Selezionare "Is Transparent" nei Lightmap Parameters in uso.

Come detto, la soluzione non è particolarmente elegante, anche perché finisce con l'illuminare in maniera eccessiva le parti che dovrebbero essere occluse (con la conseguenza di dover agire sui parametri che regolano l'intensità dell'illuminazione indiretta, nella fonte di luce o nella scheda Lighting – Scene) ma, se si desidera utilizzare Precomputed Realtime in una situazione simile, probabilmente sarà necessario adottarla.



L'illuminazione indiretta generata con Precomputed Realtime.

Dopo aver effettuato un nuovo Baking, potremo variare i parametri della fonte di luce, come l'intensità o anche il parametro Indirect Multiplier, che regola l'intensità dell'illuminazione indiretta fornita dalla fonte di luce in uso.

Parlando di parametri per il calcolo dell'illuminazione indiretta, approfittiamone per esaminare i parametri della sezione Precomputed Realtime GI dei Lightmap Parameters:

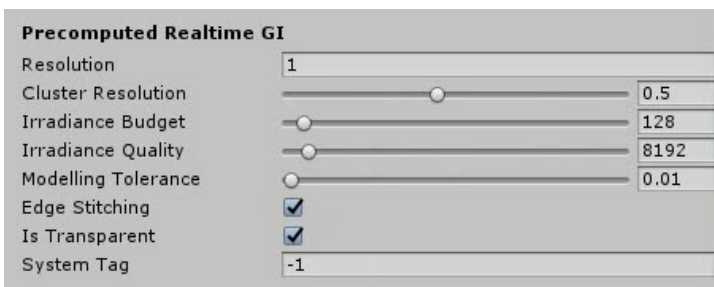
- **Resolution:** un valore, espresso in texel per unità della scena, che indica quanti texel della Lightmap riservare per ogni “metro

quadro” (o, meglio, “unità quadrata”) di superficie; il valore è MOLTO più basso di quello utilizzato per la Lightmap Resolution del metodo Baked GI, perché in Precomputed memorizziamo solo l'illuminazione indiretta nelle mappe, non l'illuminazione diretta o le ombre; il valore di default è 1 e generalmente va bene per molte scene, ma può diminuire a 0.5 o anche 0.1 per scene ampie e terreni, oppure aumentare a 2-3 per scene piccole, interni (come nel nostro caso) o comunque scene con molti dettagli;

- **Cluster Resolution:** tecnicamente, è il rapporto tra la risoluzione dei Cluster e la risoluzione delle Lightmap che verranno generate dopo aver calcolato l'illuminazione indiretta che rimbalzerà, appunto, tra i vari Cluster; in genere, il valore di default 0.5 andrà più che bene;
- **Irradiance Budget:** in un certo senso, una misura della qualità dell'illuminazione indiretta, in quanto relativa alla memoria da utilizzare per ciascun Texel della Lightmap; si consiglia di non esagerare con tale valore, perché avrà degli effetti anche sulla qualità di memoria utilizzata anche durante l'esecuzione;
- **Irradiance Quality:** il numero di raggi da utilizzare per calcolare quali Cluster della scena influenzeranno un texel della Lightmap generata, quindi ovviamente a valori maggiori si avranno risultati più accurati e, in questo caso, la memoria verrà richiesta solo nella fase di precalcolo, non durante l'esecuzione (a differenza di quanto avviene con Irradiance Budget, quindi; i due parametri vanno comunque in combinazione: Irradiance Budget indica la granularità dell'informazione che arriva dai raggi di Irradiance Quality, quindi se si aumenta Quality ma si mantiene basso Budget si avrà comunque un risultato modesto);
- **Modelling Tolerance:** se in una geometria sono presenti delle aperture (“gaps”, letteralmente “salti”), i raggi luminosi dell'illuminazione indiretta penetreranno al loro interno solo se l'ampiezza di tali aperture sarà maggiore del valore espresso con questo parametro (per cui i raggi non entreranno in fori o aperture di una geometria più piccoli di tale valore);
- **Edge Stitching:** si assicura che le varie “isole” (per le varie parti delle superfici) delle Lightmap siano connesse tra loro,

nella mappa, in modo da evitare la visualizzazione di “scuciture” sulle superfici; si consiglia di mantenerlo attivo, come di default;

- **Is Transparent:** già descritto;
- **System Tag:** qui per Systems intendiamo quelli generati da Enlighten, descritti e visti nella scena in un capitolo precedente; Enlighten suddivide automaticamente la scena in Systems, per cui il parametro di default -1 in genere va più che bene, ma è possibile forzare la disposizione di certi elementi su Systems diversi utilizzando tale parametro.



La sezione Precomputed Realtime GI in Lightmap Parameters.

Tornando alla scheda Lighting – Scene, notiamo che adesso abbiamo pochi parametri nella sezione Lightmapping Settings, perché sono stati disabilitati quelli relativi a Baked Global Illumination; tutti i parametri di questa scheda sono stati descritti nel capitolo 9.

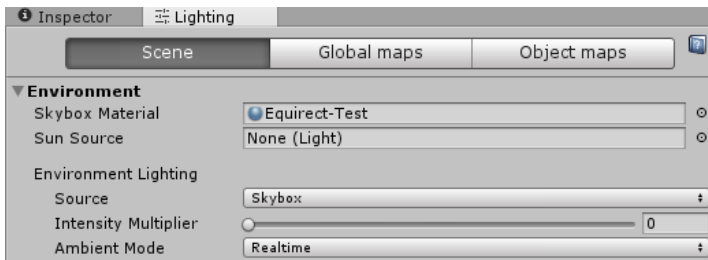
CAPITOLO 12

Luce ambientale in Real Time. Reflection Source

Luce ambientale in GI

La luce ambientale, presente anche nelle versioni precedenti di Unity, beneficia delle novità introdotte dai due sistemi di illuminazione globale Baked GI e Precomputed Realtime GI in quanto, se uno di questi due sistemi è attivato nella scheda Lighting – Scene, adesso verranno calcolate le occlusioni e l'illuminazione ambientale verrà modulata di conseguenza, per cui non si avrà più un'illuminazione costante e uniforme su tutti gli oggetti della scena, ma un effetto più realistico.

In particolare, le occlusioni verranno tenute in considerazione anche per l'illuminazione ambientale, utilizzando Precomputed Realtime GI, per gli oggetti contrassegnati come Static.



I parametri dell'illuminazione ambientale in Lighting - Scene, sezione Environment.

I parametri dell'illuminazione ambientale si trovano in Lighting – Scene – Environment Settings e, come detto nel capitolo relativo

all'illuminazione ambientale statica, consentono di definire la sorgente luminosa (tra Solid Color, Gradient e Skybox), l'intensità di tale illuminazione (Intensity) e la modalità di calcolo degli effetti di Global Illumination (Realtime o Baked).

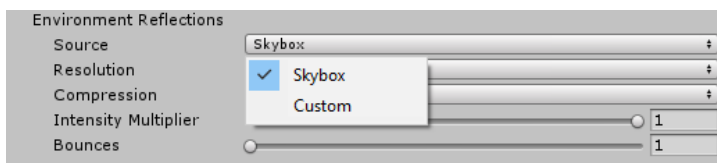
Reflection Source

Un altro parametro che, di default, fa uso di Skybox (e che va definito nella sezione Scene di Lighting) è Reflection Source, che si riferisce alla sorgente dei riflessi speculari per gli oggetti presenti nella scena.

Gli oggetti possono infatti far uso di riflessi speculari precalcolati dell'ambiente circostante, memorizzati come Textures di tipo Cubemaps (con 6 lati, come una Skybox), ovvero una “fotografia” o dello sfondo dell'ambiente virtuale o della scena vista da un dato punto nello spazio.

Tale punto è la posizione di un particolare oggetto detto Reflection Probe: una “sonda”, nel senso di un oggetto in grado di rilevare informazioni sull'ambiente circostante.

La “fotografia virtuale” dell'ambiente viene quindi miscelata agli ombreggiatori dei Materials per restituire dei riflessi speculari credibili, pur non implementando un sistema di raytracing completo in tempo reale sui vari elementi.



Parametri di riflessione ambientale e tipi di Reflection Source possibili: Skybox o una Cubemap personalizzata.

Il parametro Reflection Source di Lighting – Scene ci consente quindi di impostare l'utilizzo dello sfondo dell'ambiente virtuale (Skybox, selezionato di default), a meno di non inserire una Reflection Probe in

un punto specifico della scena, oppure quello di una Cubemap personalizzata (da inserire come Asset all'interno del progetto).

Gli altri due parametri relativi all'effetto di riflessione speculare, Bounces ed Intensity, consentono di definire ovviamente il numero di rimbalzi da calcolare per i raggi luminosi di tali riflessi e l'intensità dell'effetto.

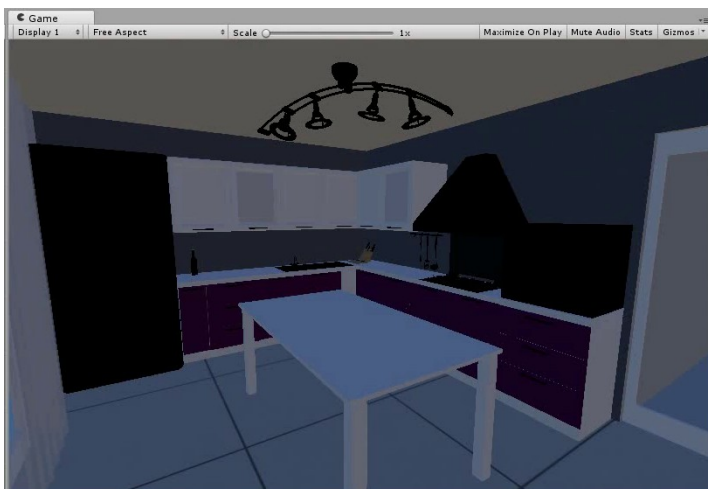
Ovviamente, la Reflection Source ambientale è indicata soprattutto per ambienti all'aperto, mentre non lo è per ambienti al chiuso, come la scena da noi utilizzata, dove è necessario calcolare i riflessi localmente, da un punto posto all'interno della stanza; vedremo come risolvere questo problema nel prossimo capitolo, parlando delle Reflection Probes.

In pratica: esempio pratico di luce ambientale in GI nel nostro progetto

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

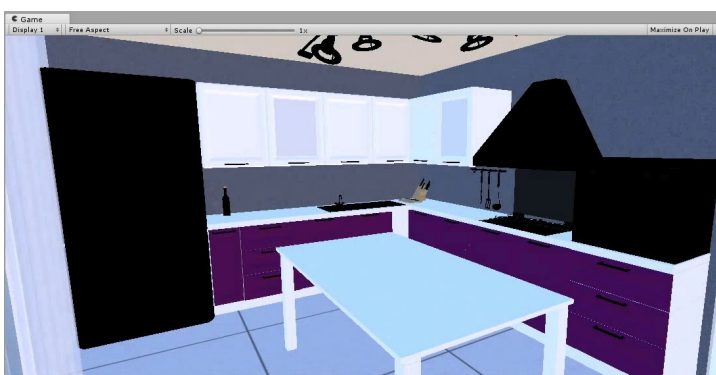
Togliamo eventuali fonti di luce standard presenti nella scena, deseleggiamo entrambe le caselle dell'illuminazione GI (Baked e Precomputed) in Lighting – Scene ed attiviamo la casella Auto Generate, in modo da far calcolare a Unity l'illuminazione indiretta... che, in realtà, in questo momento sarà nulla, visto che abbiamo deselezionato le due modalità, ma è corretto così: partiamo da una scena con illuminazione ambientale non real-time per analizzare le differenze introdotte da Precomputed Realtime per tale sistema.

Per evitare “distrazioni” con i riflessi ambientali, portiamo Intensity Multiplier di Environment Reflection a 0.0, impostando invece Environment Lighting Source su Skybox ed impostandone l'intensità a 1.0, per analizzare l'illuminazione ambientale uniforme nella scena.



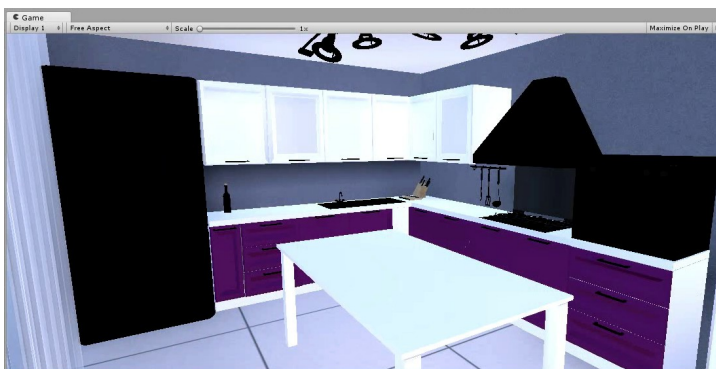
La situazione iniziale, con la sola illuminazione ambientale.

L'illuminazione così generata è uniforme, senza ombreggiature ed effetti di occlusione tra gli oggetti. Aumentando il valore di Intensity aumenterà la luminosità delle superfici, ancora una volta in maniera uniforme.



L'illuminazione ambientale di base è uniforme.

Attivando adesso Realtime GI in Lighting – Scene, verranno introdotti degli effetti di illuminazione indiretta tra le superfici vicine, che contribuiranno così a rendere meno piatta l'illuminazione della scena.



L'illuminazione ambientale dopo aver attivato Precomputed Realtime.

Anche in questo modo, comunque, la sola illuminazione ambientale (senza cioè l'utilizzo di altre fonti di luce o altri effetti) resta una soluzione valida per un determinato tipo di progetti, ad esempio per un ottenere un look stilizzato o per ridurre i requisiti del sistema di destinazione (quando si realizza un gioco per un dispositivo mobile non di fascia alta, ad esempio), oppure per realizzare anteprime di posizioni, materiali e textures degli oggetti in una scena prima di passare ad altri sistemi di illuminazione.

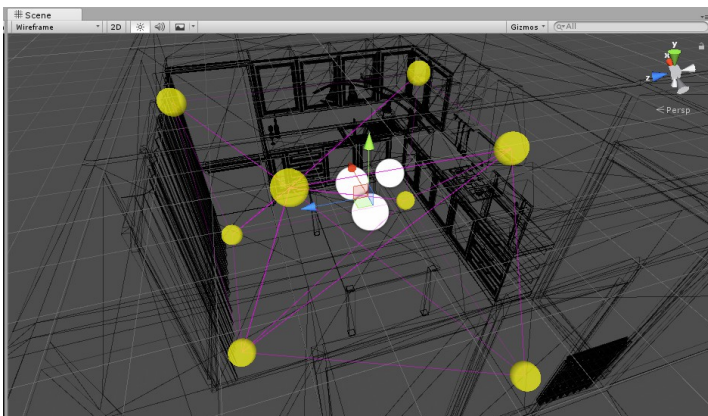
CAPITOLO 13

Light Probes, Reflection Probes: illuminazione e shading su oggetti dinamici

A cosa servono le Light Probes

Dal momento che i sistemi Baked e Precomputed Realtime della Global Illumination di Unity calcolano Lightmaps ed altri effetti di illuminazione solo per gli oggetti statici, per poter ombreggiare in tempo reale anche gli oggetti dinamici è necessario utilizzare una tecnica che fa uso di particolari oggetti chiamati Light Probes.

Le Light Probes sono “sfere virtuali” che, inserite in una scena 3D, raccolgono informazioni su colori e illuminazione lì presenti.



Le Light Probes in Unity 5.

L'interpolazione tra i valori raccolti da più Light Probes viene quindi calcolata velocemente da Unity e passata agli oggetti che si trovano all'interno del volume di influenza di queste sonde (Probes), che utilizzeranno tali valori interpolati per ombreggiare le loro superfici in maniera (relativamente) coerente con l'illuminazione presente nell'ambiente circostante.

Si tratta, quindi, di un'approssimazione: non potendo implementare il Ray Tracing su certi sistemi hardware o software, si realizza un campionamento delle informazioni e si passa questo “riassunto” agli oggetti, che miscelano queste informazioni di illuminazione e ombreggiature con quelle proprie, ottenendo un risultato finale che è comunque accettabile.

Creare e disporre le Light Probes: Light Probes Group

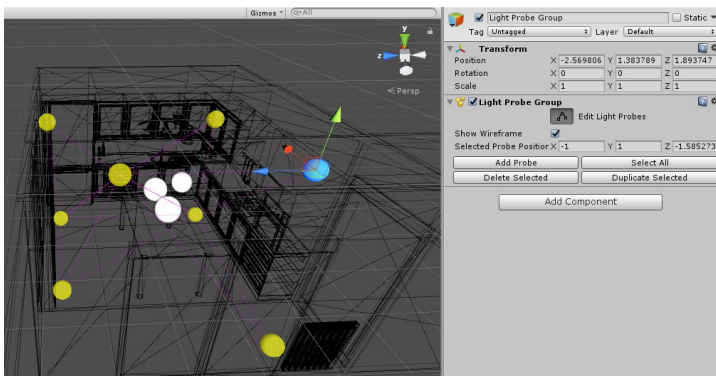
Per inserire dei Light Probes nella scena è necessario creare un Light Probe Group, che è un oggetto di gioco a tutti gli effetti e che va quindi inserito nella scena da Game Object – Light – Light Probe Group.

Di default, un Light Group appena creato avrà 6 Light Probes, che possono essere selezionate e spostate per individuare punti particolari di una scena (ad esempio, zone con molte occlusioni, porte, finestre, ecc..., ovvero tutti i punti ove l'illuminazione può variare anche in maniera notevole rispetto ad un altro punto).

Nell'Inspector del Light Probes Group sono presenti dei pulsanti (accessibili dopo aver cliccato su Edit Light Probes, che ci consentirà appunto di modificare le singole sonde all'interno di un gruppo) che consentono di aggiungere o rimuovere sonde, da posizionare poi nella scena virtuale:

- Add Probe (aggiungi una sonda-sfera al gruppo);
- Delete Selected (elimina sonda o sonde selezionate);
- Select All (seleziona tutte le sonde del gruppo);

- Duplicate Selected (crea nuove sonde per duplicazione dalla selezione corrente; utile ad esempio se ci sono tre sonde disposte in un certo modo e ce ne servono altre tre, con la stessa disposizione, in un altro punto della scena).



Trasformare un Light Probe Group.

Baking delle Light Probes

Le sonde devono passare da un loro processo di Baking, prima di avviare il gioco, perché in effetti si tratta di oggetti statici che recuperano l'illuminazione globale: la passeranno, poi, agli oggetti dinamici a runtime, ma prima devono precalcolarla.

Ciò può essere fatto, come sappiamo, impostando Auto Generate in Lighting – Scene, oppure manualmente con un click su Generate Lighting.

In pratica: esempio di utilizzo delle Light Probes nel nostro progetto

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

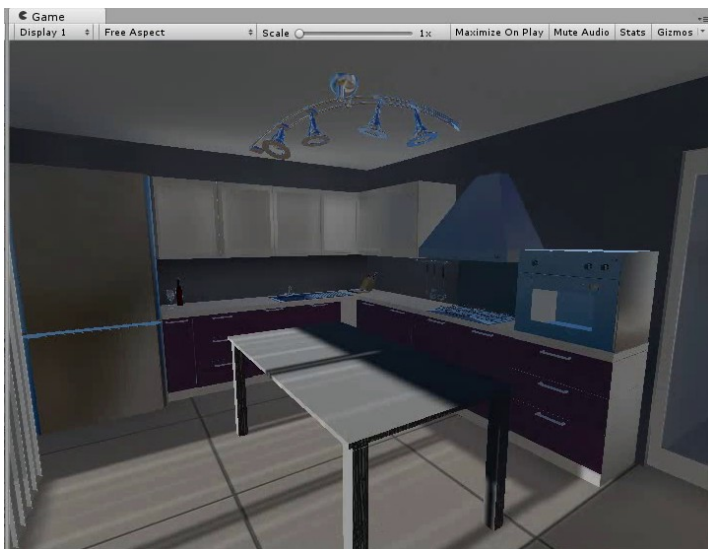
Per provare il funzionamento delle Light Probes nella nostra scena d'esempio, fornita con il pacchetto Assets di questa guida, possiamo utilizzare il tavolo: un elemento posto al centro della stanza che potrebbe essere soggetto a spostamenti in tempo reale, ad esempio per visualizzare un ambiente in maniera interattiva modificando in realtime questo elemento; ovviamente, il discorso va esteso a tutti gli oggetti dinamici di un progetto, come ad esempio i personaggi di un videogioco: i concetti relativi all'utilizzo delle Light Probes valgono in generale.

Configuriamo il progetto in questo modo:

1. in Lighting – Scene, portiamo a 0.0 l'intensità dell'illuminazione ambientale, in modo da concentrarci solo sull'illuminazione diretta e indiretta proveniente da una fonte di luce standard; i riflessi ambientali, invece, possono essere lasciati a 1.0 (valore di default);
2. selezioniamo solo la modalità Precomputed Realtime GI, lasciando la modalità Baked deselezionata;
3. attiviamo Auto Generate in fondo alla scheda, in modo da far effettuare il precalcolo dell'illuminazione a Unity in maniera automatica;
4. assicuriamoci che la casella “Is Transparent” sia attiva per le impostazioni Lightmapping Settings di questo progetto, come discusso nei capitoli precedenti;
5. inseriamo una fonte di luce Direzionale, Real-Time, con ombre Soft Shadows e valori Rotation XY a 150.0 e 0.0 rispettivamente, in modo da illuminare il tavolo;
6. selezioniamo l'oggetto KITCHEN, in cima alla gerarchia di geometrie, assicurandoci che tutti gli oggetti siano STATIC (tramite la Checkbox nell'Inspector), quindi selezioniamo solo

l'oggetto Table e, per tale elemento, deselezioniamo la casella Static, rendendolo così dinamico.

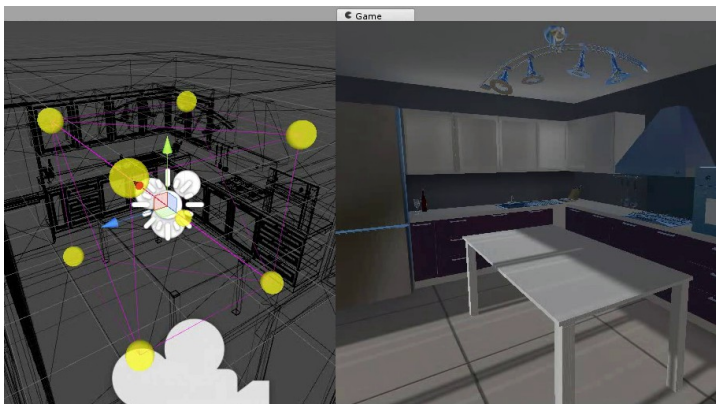
Unity terrà subito conto dell'ultima operazione, rimuovendo il tavolo dal precalcolo dell'illuminazione indiretta e, di fatto, rendendolo scuro nelle parti non direttamente illuminate dalla fonte di luce Directional Light: tali parti, infatti, non usufruiranno più dell'illuminazione indiretta precalcolata.



La situazione con Precomputed Realtime (no Baked GI) e il tavolo "dinamico" (non Static).

Aggiungiamo ora un oggetto di tipo Light Probes Group nella scena, con click destro in Hierarchy e Light – Light Probes Group.

L'oggetto verrà inserito al centro della scena e gli effetti, dopo il precalcolo effettuato automaticamente da Unity, saranno subito visibili: trovandosi nella zona di influenza delle sonde create con Light Probes Groups, il tavolo – o un oggetto dinamico in generale – verrà illuminato in tempo reale anche con illuminazione di rimbalzo.



Dopo aver inserito un Light Probe Group nella scena, il tavolo verrà illuminato con illuminazione indiretta.

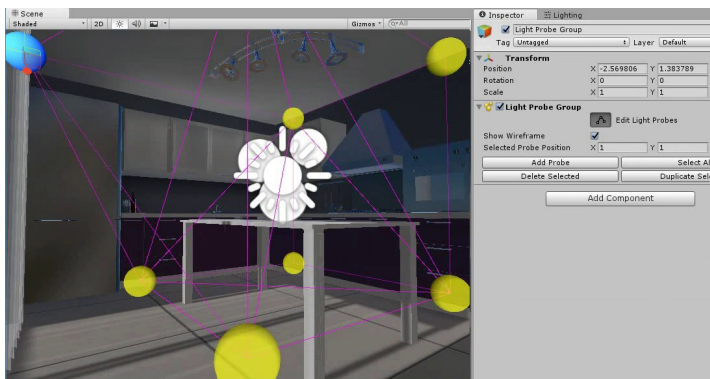
L'oggetto, comunque, NON produrrà effetti di illuminazione indiretta in real-time sugli altri elementi della stanza, ma potrà solo riceverla, grazie alle Light Probes precalcolate.



Un oggetto dinamico illuminato con le Light Probes potrà solo ricevere l'illuminazione indiretta precalcolata, non proiettarla in tempo reale sugli altri oggetti.

Le Light Probes possono essere utili anche con oggetti che tecnicamente non dovrebbero essere dinamici, ma che – per dimensioni ridotte o posizione nella scena – probabilmente darebbero un contributo quasi nullo all'illuminazione indiretta di altri oggetti: in questo caso, per ridurre il tempo di precalcolo senza intaccare in maniera sensibile la qualità del risultato a real-time, si può scegliere di rendere dinamici tali oggetti e far gestire alle Light Probes la loro illuminazione indiretta.

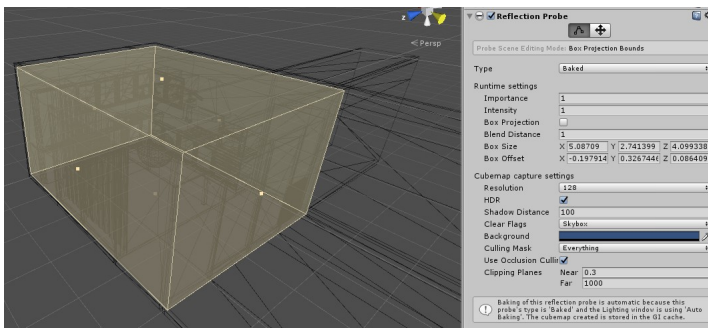
Usciamo dalla modalità Play per individuare i comandi che ci consentiranno di trasformare le sonde del gruppo; in particolare, con l'oggetto Light Probes Group selezionato ed un click sulla casella “Edit Light Probes” nell'Inspector di tale oggetto, potremo accedere alle singole sonde luminose, aggiungendone di nuove o rimuovendo quelle superflue, posizionandole in particolare lì dove l'illuminazione cambia in modo particolare, ad esempio in prossimità delle finestre e delle porte o sotto altri oggetti.



Modifica (Editing) delle Light Probes.

Riflettere l'ambiente su oggetti statici e dinamici: Reflection Probes

Finora abbiamo utilizzato lo Skybox di scena, ovvero lo sfondo dell'ambiente virtuale, per generare i riflessi speculari sugli oggetti, con effetti evidenti in particolare per gli oggetti metallici (frigorifero, forno, lampadario, ...). In Unity 5 è possibile definire dei particolari oggetti detti Reflection Probes: si tratta di sonde, come le Light Probes, che “scattano una fotografia” dell'ambiente che hanno intorno e memorizzano questa informazione per applicare i riflessi speculari agli oggetti che si trovano all'interno del loro campo d'azione, definito mediante un parallelepipedo che possiamo ridimensionare a piacere nelle tre dimensioni.



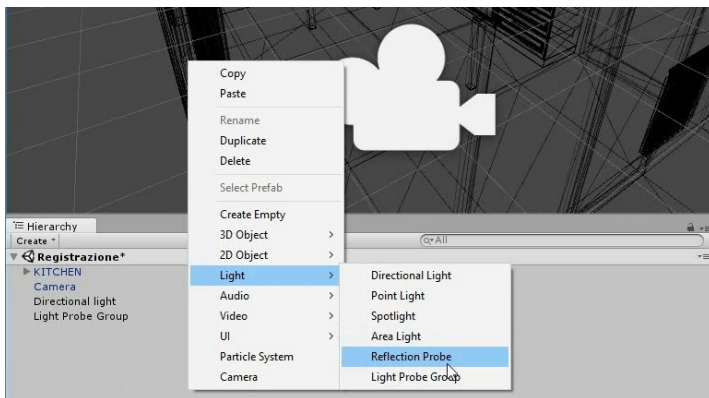
Trasformare le Reflection Probes, definendo in particolare la posizione (dalla quale verrà "scattata la fotografia" dell'ambiente circostante) e il parallelepipedo che definirà l'area di influenza della sonda (ombreggiando gli oggetti che si troveranno al suo interno).

Le Reflection Probes possono essere Baked o Real-Time; nel primo caso (Baked), i riflessi della sonda verranno precalcolati prima dell'esecuzione del progetto, per cui la Texture dei riflessi (detta Cube Map) non potrà essere modificata in tempo reale e non mostrerà quindi i riflessi di oggetti dinamici; nel caso delle sonde Real-Time, invece, la “fotografia dell'ambiente” verrà aggiornata durante l'esecuzione del programma e, quindi, sarà possibile ottenere e visualizzare la riflessione di oggetti dinamici presenti nell'ambiente virtuale.

In pratica: esempio di utilizzo delle Reflection Probes nel nostro progetto

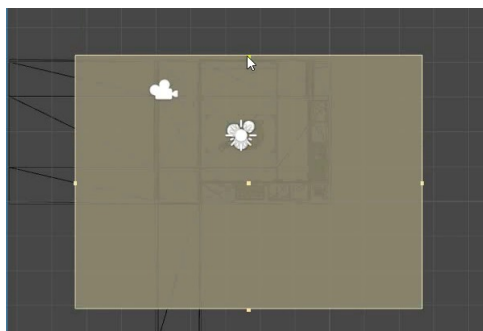
Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Le Reflection Probes, al pari delle Light Probes, andrebbero inserite nei punti dove l'ambiente circostante cambia notevolmente, perché ovviamente cambieranno anche gli elementi da riflettere: in un interno, ci vorrà ALMENO una Reflection Probe per ogni stanza, mentre ad esempio in prossimità di una galleria in un gioco d'auto ce ne vorrà una posizionata prima dell'ingresso della galleria, almeno una all'interno (a seconda della struttura e degli elementi presenti nella galleria) e un'altra all'uscita, per far riflettere correttamente l'ambiente esterno all'oggetto, altrimenti si finirà nell'errore fatto finora con gli oggetti in metallo all'interno della stanza, che riflettevano lo Skybox dell'ambiente esterno!



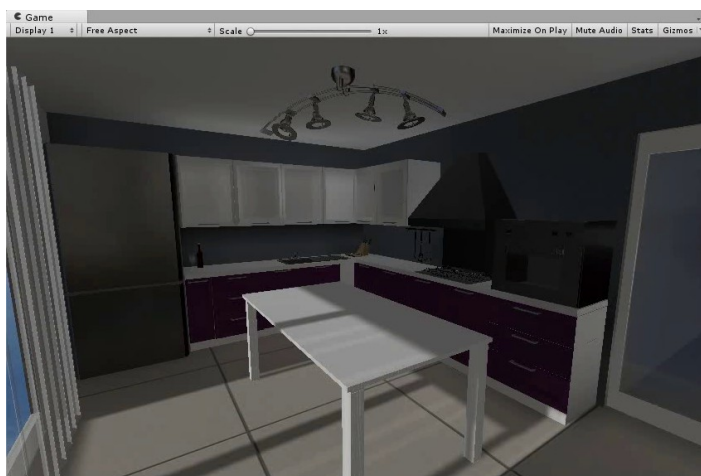
Inserimento di una Reflection Probe nella scena, con click destro in Hierarchy, quindi Light - Reflection Probe.

Per aggiungere una Reflection Probe all'interno della nostra scena d'esempio, fornita con il pacchetto Assets di questa guida, facciamo click destro nella scheda Hierarchy, quindi scegliamo Light – Reflection Probe.



Editing della Reflection Probe.

Con il precalcolo dell'illuminazione in maniera automatica, in Lighting – Scene, l'effetto della Reflection Probe sarà subito evidente sugli oggetti metallici che si troveranno all'interno del suo parallelepipedo; ricordiamo, infatti, che la “fotografia” dell'ambiente virtuale viene scattata dal punto in cui si trova la sonda, mentre il parallelepipedo indica il volume all'interno del quale saranno presenti i suoi effetti, per cui solo gli oggetti presenti in tale parallelepipedo verranno ombreggiati con la Reflection Probe.



Effetti della presenza della Reflection Probe subito visibili, in particolare sugli oggetti con superfici metalliche.

È possibile visualizzare gli effetti della Reflection Probe anche agendo sull'omonima checkbox all'interno dell'Inspector dell'oggetto; l'anteprima della Texture Cube Map generata dalla Reflection Probe, per agire sui riflessi speculari degli oggetti, è visibile nel suo Inspector.



L'anteprima della Reflection Probe, nella sezione Preview dell'Inspector di tale oggetto.

Se non intendiamo aggiungere elementi dinamici nell'ambiente, possiamo impostare la modalità della Reflection Probe a Baked, nel suo Inspector, altrimenti dovremo passare alla modalità Real-Time.

Per modificare forma e dimensioni del parallelepipedo che definisce l'area di influenza della sonda, cliccare sull'icona Box Projection Bounds nell'Inspector della sonda, quindi cliccare e spostare i quadratini che rappresentano il centro di ciascuna faccia, nel parallelepipedo visibile nella scheda Scene.

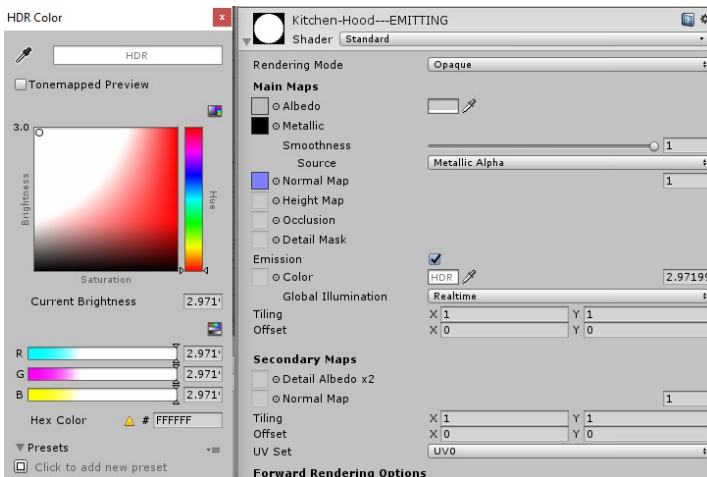
CAPITOLO 14

Materiali emissivi e real-time GI

Emissive Materials e Lightmap Static

I Materials degli oggetti hanno a disposizione la proprietà Emissive, per cui le geometrie possono diventare delle fonti di luce all'interno della scena, con colori e intensità proprie.

Per poter partecipare alla simulazione di Illuminazione Globale, sia le mesh emettitrici che gli oggetti di scena che dovranno ricevere la loro illuminazione devono essere contrassegnati come “Lightmap Static”, nella scheda Object della finestra Lighting per gli oggetti selezionati.



Un Material con il parametro Emission attivato e il selettore del suo colore aperto a sinistra (dopo aver cliccato sulla Color Box accanto ad Emission - Color).

Inspector Static e problemi a runtime con Emissive Non Static

Anche nell'Inspector è presente un parametro “Static” e, selezionando tale casella, Unity selezionerà automaticamente Lightmap Static; tuttavia, “Static” di Inspector si riferisce a TUTTE le proprietà dell'oggetto che possono essere statiche o dinamiche, incluse le animazioni e gli spostamenti in tempo reale. Attivando quindi Static in Inspector, un oggetto diventerà fisicamente immobile, immutabile.

Va fatta una considerazione anche per gli oggetti Lightmap Static ma non “Inspector Static”: in questo caso, nella casella Static dell'Inspector vedremo il segno -, anziché il segno di spunta, ad indicare che l'oggetto è solo parzialmente statico (e, nello specifico, lo è relativamente alla Light Map).

L'oggetto potrà quindi essere spostato, ma a questo punto si potrebbe verificare un problema con l'illuminazione realizzata in maniera Lightmap Static con i materiali emissivi: tale illuminazione, infatti, NON verrà aggiornata; ciò potrebbe portare, chiaramente, a rappresentazioni incoerenti a runtime.

Le Light Probes e gli oggetti dinamici da illuminare tramite Emissive

Per poter illuminare gli oggetti dinamici (non Lightmap Static) tramite oggetti “emissivi” (che, lo ricordiamo, devono essere di tipo Lightmap Static per partecipare all'illuminazione globale), è necessario far ricorso a particolari Componenti detti Light Probes, che dovranno essere utilizzati proprio dagli oggetti dinamici. Abbiamo parlato delle Light Probes nel capitolo precedente; utilizzando le Light Probes, sarà possibile illuminare gli oggetti dinamici anche con gli oggetti provvisti di materiali emettitori.

Oggetti emissivi in movimento

Abbiamo descritto quindi l'interazione tra oggetti Emissive e oggetti statici della scena (Lightmap Static, per entrambi) e abbiamo chiarito che per illuminare gli oggetti dinamici con oggetti Emissivi statici è necessario utilizzare i Light Probes, visti nel capitolo precedente; resta quindi il caso della realizzazione di oggetti dinamici (in movimento) dotati di Materials Emissive.

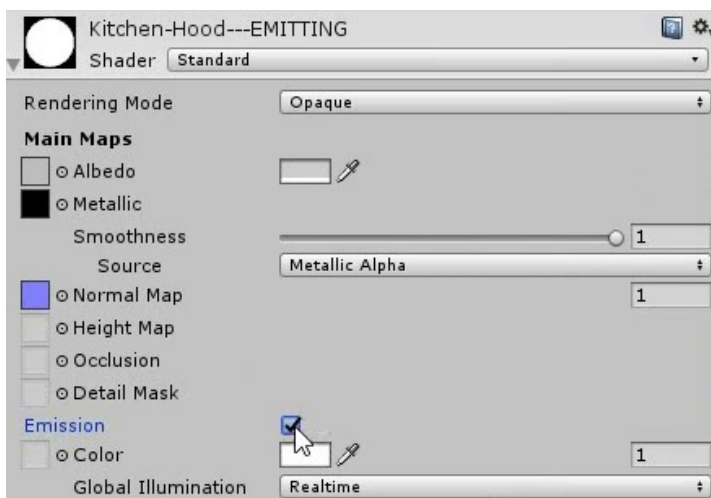
Purtroppo, almeno per le versioni correnti di Unity il manuale parla chiaro: “Emission” consente ai soli oggetti statici di emettere luce e illuminare gli altri oggetti nella scena. Un materiale emissivo applicato ad una geometria dinamica NON genererà illuminazione nella scena, ma si limiterà ad essere – come avviene anche nel caso di Emissive Static – auto-illuminato, come un elemento fosforescente o, se vogliamo, “evidenziato”.

In pratica: un materiale emissivo Real-Time animato, con script, nel nostro progetto

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

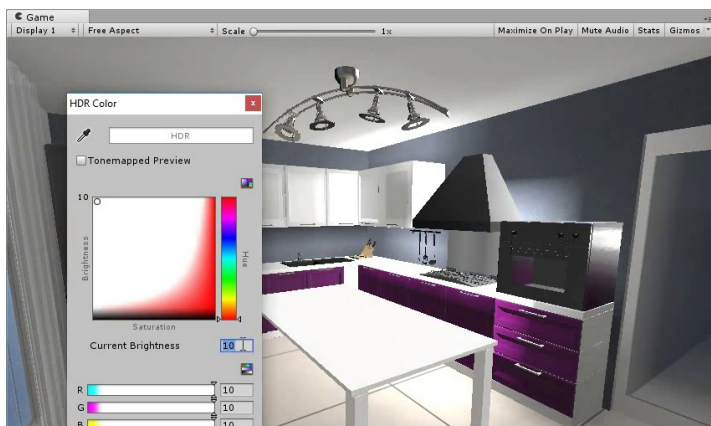
Per provare un materiale Emissivo all'interno della nostra scena d'esempio, fornita il pacchetto Assets di questa guida, utilizziamo uno dei sottomateriali dell'oggetto Kitchen-Hood (la cappa), in particolare “Emitting”.

Il campo Emission di tale sottomateriale, accessibile dall'Inspector dell'oggetto Kitchen-Hood, sarà disabilitato di default; dopo averlo attivato, potremo impostare il colore e l'intensità luminosa della superficie emissivo; quest'ultimo aspetto potrà essere definito tramite il parametro Current Brightness, nella casella del colore, e potrà avere valori anche molto maggiori di 1.0.



Attivare la casella Emission di un materiale per renderlo un emettitore luminoso.

Il parametro Global Illumination potrà essere impostato su Baked o Realtime a seconda del sistema di illuminazione globale da utilizzare (in Lighting – Scene).



Variare colore e intensità (Current Brightness) dell'emissione luminosa, cliccando sulla Color Box accanto ad Emission, nella scheda del Material.

Rendiamo ora dinamica l'emissione luminosa nella scena tramite un semplice script che possiamo creare in questo modo: selezioniamo l'oggetto Kitchen-Hood, quindi clicchiamo su Add Component in fondo al suo Inspector e creiamo uno script di tipo JavaScript chiamato Dimmer, cliccando su “Create and Add”.

Facciamo doppio click sul componente appena creato per aprire l'editor degli script (che nel mio caso è MonoDevelop, nativo di Unity, selezionato da Edit – Preferences – External Tools – External Script Editor) e definiamo il contenuto dello script in questo modo (salvando lo script subito dopo aver incollato questo contenuto, in modo da salvare le modifiche e far aggiornare a Unity lo script nell'Inspector):

```
#pragma strict

var color : Color = new Color(0.0F, 0.0F,
0.0F, 1.0F);

public var material : Material;

function Start () {
}

function Update () {
    color.r = color.g = color.b = 3 *
    Mathf.Abs(Mathf.Sin(Time.time));
    material.SetColor("_EmissionColor",
    color);
    RenderExtensions.UpdateGIMaterials(gameObject.GetComponent(Renderer));
}
```

In pratica, abbiamo fornito allo script una variabile di tipo color, che conterrà appunto il colore da assegnare al nostro materiale; abbiamo definito, inoltre, una variabile globale – visibile nell'Inspector – di tipo Material, che utilizzeremo per legare il Material con emissione luminosa allo script, in questo modo: nell'Inspector, clicchiamo e trasciniamo il

Material Kitchen-Hood---Emitting nello slot Material dello script Dimmer.

All'interno della funzione Update, che viene eseguita automaticamente da Unity ad ogni frame durante l'esecuzione del progetto, abbiamo tre istruzioni:

- con la prima, assegniamo un valore uguale ai tre canali rosso, verde e blu della variabile color, ottenendo di fatto un grigio, la cui intensità varierà da 0 a 3, perché deriverà da $3 * \text{il valore assoluto del seno dell'istante temporale}$; poiché il seno di un valore può variare da -1 a 1 e noi lo stiamo passando in valore assoluto e moltiplicando per 3, il valore per i tre canali RGB del colore varierà da 0.0 a 3.0;
- con la seconda istruzione assegniamo il colore appena calcolato al colore Emission del Material, che abbiamo legato poco fa allo script tramite trascinamento nell'Inspector;
- con la terza istruzione diciamo a Unity di aggiornare i materiali che partecipano alla Global Illumination per l'oggetto che fa uso di questo script.

Queste tre operazioni verranno eseguite automaticamente, come detto, ad ogni frame dell'esecuzione del progetto.

Salviamo le modifiche nell'Editor e clicchiamo sul pulsante Play per osservare il risultato in tempo reale.

CAPITOLO 15

Camera HDR e Post-Processing Stack***LDR vs HDR; Tonemapping***

Nei programmi di grafica e di rendering i valori di intensità luminosa (sia per la scala di grigi che per i vari canali-colore) vengono codificati, in genere, utilizzando 8 bit di memoria; vengono poi espressi con valori nel range 0.0 – 1.0 o nel range 0 – 255.

8 bit per la scala di grigi o per ciascun canale-colore costituiscono una quantità di memoria (e, di conseguenza, un grado di precisione) sufficiente per una buona rappresentazione delle scene... buona, ma non ottima: il range dei valori di intensità del mondo reale da rappresentare è molto, molto più ampio; non solo: per via della percezione che il nostro cervello ha delle intensità luminose (seguendo una scala logaritmica, non lineare, oltre a certi “scherzetti” che può farci riguardo la percezione del colore o dell'intensità luminosa in certi contesti), sono necessarie più informazioni in certe parti della scala piuttosto che in altre, per cui la codifica a 8 bit, detta LDR (Low Dynamic Range), non è sufficiente.

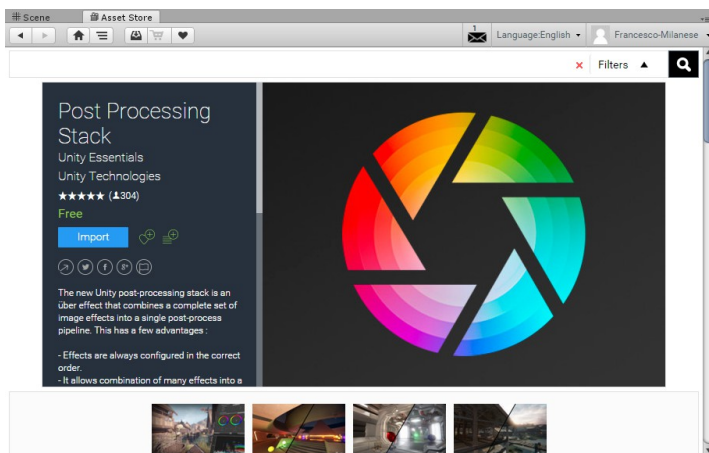
Per questi motivi, molti programmi consentono di lavorare utilizzando un range di valori più ampio, detto HDR (High Dynamic Range), memorizzando i dati in formati a 16 o 32 bit e consentendoci di effettuare operazioni particolari in tale range.

Alla fine, tali dati andranno comunque riportati al range LDR (perché è in questo range che lavora la maggior parte dei dispositivi di visualizzazione), ma ciò avviene dopo aver effettuato operazioni particolari di elaborazione delle immagini (con filtri ed altri strumenti,

come il Glow) nel range più ampio. La conversione da HDR a LDR è detta, in particolare, Tonemapping.

Unity consente di elaborare i dati in modalità HDR, prima di presentarli a video, consentendoci di effettuare operazioni particolari che, usufruendo della maggior quantità di dati a disposizione, possono essere effettuate in maniera più accurata.

Quando si fa uso della modalità HDR per una telecamera virtuale in Unity, bisognerebbe aggiungere un componente Tonemapping che, a partire da Unity 5.6, è disponibile insieme ad altri effetti nel pacchetto Post-Processing Stack, scaricabile gratuitamente dall'Asset Store; in questo modo, sarà possibile definire e personalizzare la modalità di Tonemapping utilizzata dalla telecamera in uso.



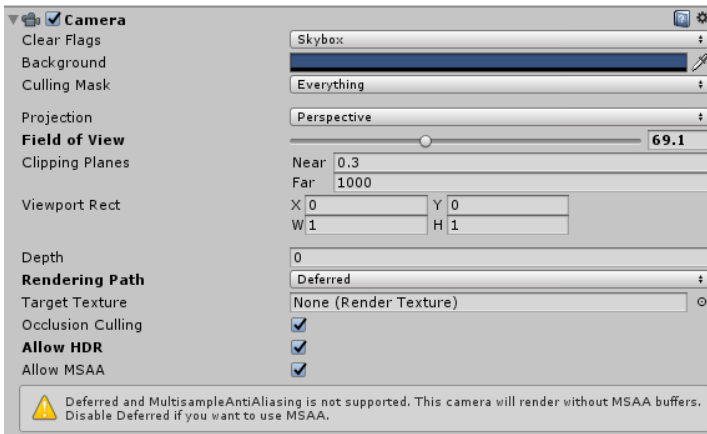
Il pacchetto Post-Processing Stack, nella finestra Asset Store.

HDR in Unity; HDR per la telecamera

L'HDR può essere utilizzato in Unity sia dalle immagini textures utilizzate come illuminanti (negli ambienti virtuali, come ad esempio per Skybox, oppure negli oggetti, come ad esempio i materiali emissivi) sia

dalla telecamera virtuale. In questo capitolo ci occupiamo del secondo caso.

Ogni telecamera virtuale di Unity mette a disposizione una casella di spunta chiamata HDR; selezionandola nell'Inspector si indicherà a Unity di elaborare l'immagine da renderizzare nel range HDR, il che ci consentirà di applicare degli effetti o filtri particolari utilizzabili in tale range prima di passare al Tonemapping, che ri-mapperà l'immagine renderizzata in LDR per mostrarla sul display o dispositivo di visualizzazione.



Il parametro Allow HDR per la telecamera virtuale; selezionarlo per usufruire al meglio degli effetti Post-Processing Stack.

Buffer e filtri HDR; note su Forward e Deferred Rendering

Quando l'opzione HDR è selezionata, quindi, l'immagine viene renderizzata in un buffer (un'area di memoria) particolare, consentendoci di elaborare i dati con filtri e altri strumenti che operano al di fuori del range LDR. Tra i vantaggi dell'utilizzo di HDR vi è il fatto che i colori non vanno perduti ("troncati" a bianco) nelle aree molto

luminose e non si ha, in genere, la formazione di “spot luminosi” nelle aree sovraesposte.

È chiaro che non tutti i supporti hardware mettono a disposizione tale buffer o comunque non sono in grado di gestire queste particolari operazioni, in quanto è richiesta molta memoria video e le operazioni sono in virgola mobile, per cui potrebbero verificarsi delle restrizioni (o tutto il procedimento potrebbe non essere supportato affatto) a seconda del dispositivo in uso.

Se si fa uso della modalità di rendering Forward, il passaggio in HDR e il relativo Tonemapping per tornare in LDR verranno effettuati solo in presenza di filtri in tale range, altrimenti il passaggio verrà ignorato (per motivi legati alle performance); al contrario, Deferred Rendering effettuerà il passaggio ad HDR ed il conseguente Tonemapping anche in assenza di effetti particolari.

Il package Post-Processing Stack e gli effetti per le telecamere

A partire da Unity 5.6, le operazioni e i filtri utilizzabili in HDR (e non solo) sono degli effetti di post-produzione da importare con il pacchetto Post-Processing Stack dall'Asset Store.

Una volta importato ed installato il pacchetto, sarà possibile aggiungere un Component di tipo “Post-Processing Behaviour” per la telecamera; tale Component conterrà un'istanza di un particolare Asset, il Post-Processing Profile, nel quale potremo specificare gli effetti da utilizzare e i loro parametri.

Il flusso di lavoro sarà quindi il seguente: nei Post-Processing Profiles sarà possibile impostare i vari effetti che si vorranno fornire alle varie telecamere virtuali; ciascuna telecamera virtuale avrà un Component Post-Processing Behaviour, nel quale andrà specificato il Profile da utilizzare: una volta effettuata questa operazione, la telecamera implementerà quegli effetti.

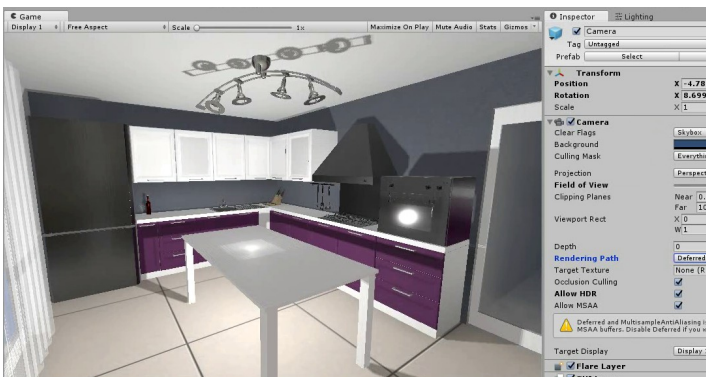
Vediamo quindi come mettere in pratica quanto detto e come specificare un paio di effetti per la nostra telecamera virtuale.

In pratica: utilizzare il pacchetto Post-Processing Stack

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Per poter provare gli effetti del pacchetto Post-Processing Stack, dobbiamo effettuare qualche operazione preliminare nella scena.

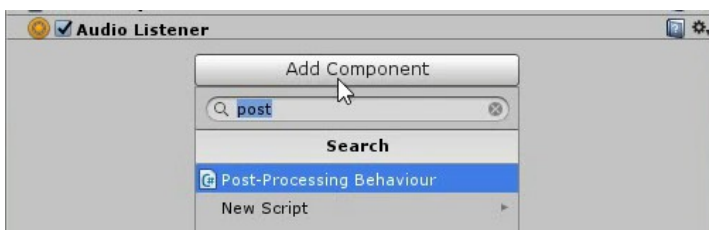
Iniziamo con l'inserire una Point Light all'interno della stanza, ad esempio in posizione centrale (che, nel mio caso, corrisponde alle coordinate XYZ -2.5, 1.0, 2.0), con una discreta intensità, ad esempio 1.5, attivando poi le Soft Shadows ed utilizzando la modalità Real-Time; eliminiamo qualsiasi altra fonte di luce eventualmente possibile ed annulliamo il contributo dell'illuminazione ambientale, in modo da utilizzare solo la Point Light come fonte di illuminazione della scena.



La scena di partenza per l'esempio pratico di questo capitolo.

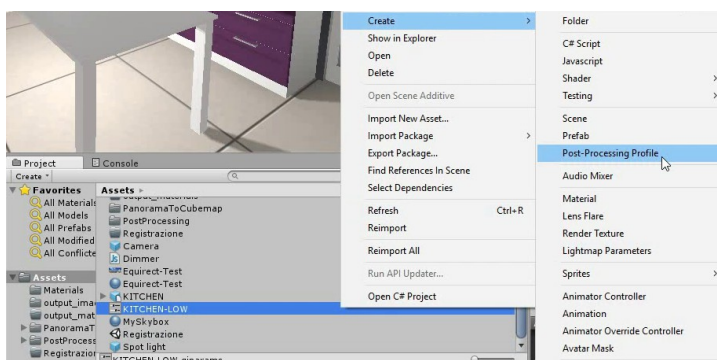
Per quanto riguarda la telecamera, nell'Inspector di tale oggetto attiviamo l'opzione Camera HDR ed impostiamo il Rendering Path Deferred.

Il pacchetto Post-Processing Stack va installato direttamente dall'Asset Store, per cui – se non l'abbiamo già fatto – colleghiamoci all'Asset Store mediante l'omonima finestra (richiamabile dal menù Window), effettuiamo l'accesso con il nostro account e procediamo con il download e l'installazione del pacchetto. A questo punto dobbiamo aggiungere un Component Post-Processing Behaviour per la telecamera, cosa che possiamo fare cliccando sul pulsante Add Component nell'Inspector della telecamera virtuale per poi scrivere “Post” nella casella di ricerca di tale elemento: Unity ci suggerirà subito Post-Processing Behaviour, per cui clicchiamo su tale voce.

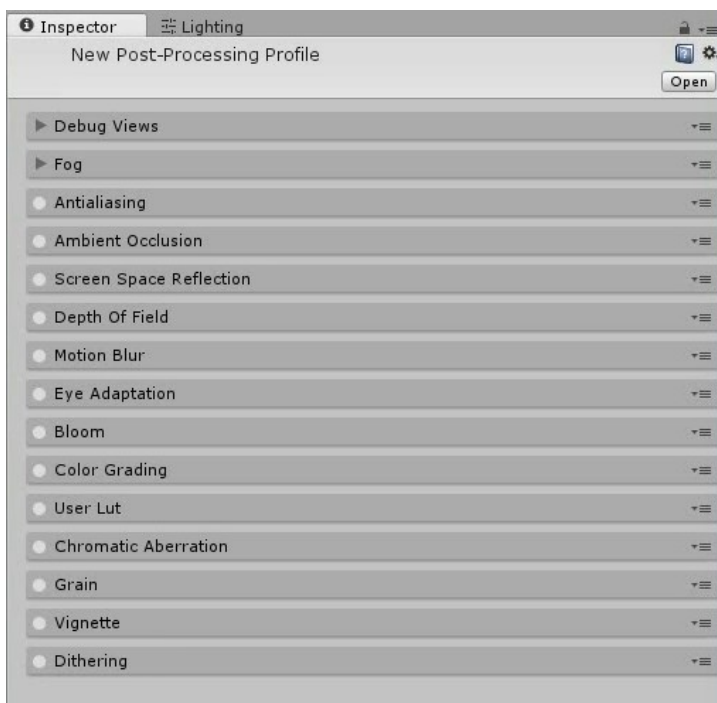


Aggiungere un componente Post-Processing Behaviour ad una telecamera virtuale.

Questo componente-script presenta un solo parametro pubblico, inizialmente vuoto: un campo chiamato Profile, che accetta un oggetto di tipo Post-Processing Profile. Un oggetto del genere contiene le informazioni sui filtri e gli effetti da implementare nelle telecamere che ne fanno uso. Ad ogni telecamera virtuale del progetto potremo quindi associare dei profili con effetti diversi, a seconda delle situazioni e dei contesti. Creiamo quindi un oggetto Post-Processing Profile con click destro nella finestra Project – Assets e scegliendo la relativa voce dal menù Create. Subito dopo aver creato il Profile, nell'Inspector verranno mostrati i vari effetti (inizialmente, tutti disattivati) che sarà possibile implementare con tale elemento... ma prima dobbiamo far utilizzare il nuovo oggetto ad almeno una telecamera virtuale!

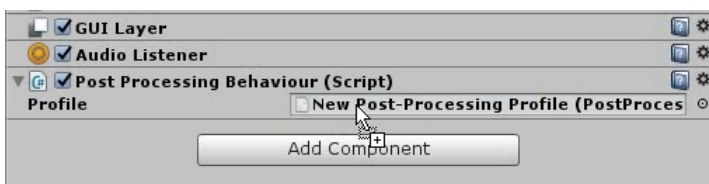


Creazione di un Post-Processing Profile.



Gli effetti disponibili in un Post-Processing Profile.

Selezioniamo quindi la telecamera principale e trasciniamo il nuovo oggetto Post-Processing Profile nel campo Profile del componente Post-Processing Stack della telecamera.



Impostare il Post-Processing Profile nel Post-Processing Behaviour della telecamera.

A questo punto possiamo selezionare Post-Processing Profile ed impostare gli effetti che ci interessano tramite le sezioni e i parametri presenti nell'Inspector per tale oggetto.

In pratica: Bloom e altri effetti

Di questo capitolo è presente online la versione video gratuita; per maggiori informazioni, visitare la pagina descrittiva di questo prodotto sul sito dell'autore, all'indirizzo indicato nell'introduzione di questa guida.

Per attivare un effetto del gruppo Post-Processing, cliccare sulla casella di spunta accanto al nome dell'effetto nell'Inspector dell'oggetto. Cliccando sulla barra del titolo di ciascun effetto si potrà aprire o chiudere la relativa scheda e, quindi, modificarne i parametri.

Questa trattazione non è dedicata agli effetti di Post-Produzione implementabili in Unity, per cui non descriveremo in dettaglio tutte le voci presenti in Post-Processing Stack, ma ci limiteremo a descrivere un paio di effetti che riguardano l'illuminazione della scena.

Ambient Occlusion scurisce le superfici che si trovano vicine ad altri oggetti o anche a sottoparti dello stesso oggetto e che, quindi, le mettono in ombra o comunque ne ostacolano l'illuminazione diretta e indiretta

(ovvero: elementi dell'ambiente che li occludono, appunto).

Ambient Occlusion è un effetto piuttosto “pesante” da gestire, per cui si consiglia di attivare l'opzione Downsampling, se questo non peggiora in maniera sensibile la qualità dell'effetto.



L'effetto Ambient Occlusion in azione nella nostra scena.

Bloom è un effetto bagliore che, sostanzialmente, “sporca” le aree che si trovano vicine a parti molto luminose (o perché emettitrici, come le superfici Emissive, o perché molto riflettenti, come ad esempio le superfici metalliche all'interno della nostra scena).

Modificando Intensity e Radius potremo definire, intuitivamente, l'ampiezza del bagliore e l'intensità dello stesso; gli altri parametri possono essere lasciati, in genere, come di default.



L'effetto Bloom in azione con le impostazioni di default.

Color Grading è l'effetto che ci consente di definire il Tonemapping, ovvero la modalità con la quale le elaborazioni effettuate da Unity nel range HDR verranno riportate al range LDR. Questo parametro influenzerà quindi, tra le altre cose, anche la resa visiva di altri effetti di post-produzione, gestiti da Post-Processing Stack.



L'effetto Color Mapping con Tonemapping ACES in azione.

In Unity 5.6 si hanno, di default, tre opzioni per tale effetto, alla voce Tonemapper: None, Neutral e Filmic (ACES).

La voce None andrebbe usata, ovviamente, quando si lavora in LDR, in quanto non effettua alcun Tonemapping.

L'opzione Neutral si limita a convertire i valori numerici dal range HDR a quello LDR, senza introdurre alcun effetto su tonalità, contrasto, saturazione o altri aspetti del colore o della luminosità: è un'operazione “neutrale”, appunto.

Il metodo Filmic si basa su un sistema di codifica chiamato ACES (Academy Color Encoding System) che, come suggerisce il nome, effettua delle operazioni sul colore, per cui in effetti si hanno delle variazioni nelle tonalità e nella saturazione dei colori; inoltre, il contrasto risulta più accentuato, rispetto agli altri metodi.

I vari effetti possono essere attivati o disattivati singolarmente; per visualizzare invece le differenze introdotte da tutti gli effetti contemporaneamente, è possibile attivare o disattivare l'intero componente Post-Processing Behaviour: in questo modo, sarà possibile valutare velocemente il contributo e gli effetti della Post-Produzione.
