



クリアな透明オブジェクトのレンダリングは、どのレンダリングエンジンにおいても最も時間のかかる作業の一つであり、コンピュータグラフィックスの分野で広く研究されています。しかし、現実世界の透明な物体はほとんどきれいではありません。それらの表面には、ホコリ、汚れ、指紋、塩、石鹼、水など、さまざまな汚染物質が存在します。これらの汚染物質は、フォトリアルなレンダリングを作成する際に考慮すべきさまざまなボリューム散乱効果を生み出します。

クリアな透明表面のレンダリング画像は、完璧すぎる印象を与えがちです。実際の写真で見ると、透明な物体にはほぼ常に表面の汚れがあり、こうした不完全さは風化した外観を生み出します。これはリアリズムを達成するために重要です。

汚染物質を滑らかな透明素材の上に重なる光学的に薄い層として考えると、理論的には2つの素材をアルファブレンドで単純に重ねることができます。しかし、実際には汚染物質ごとに光を散乱させたり減衰させたりする特有の性質があり、それによって顕著な視覚効果が生じることがあります

次のようなものがあります：

コントラスト反転：汚れた窓を通して見ると、汚れは明るい背景に対して暗く見え、暗い背景に対して明るく見える。

角度依存性：汚染物質を通過する光の減衰と透過は、カメラの視角によって変化します。

レンズのグレアとぼやけ：透明な表面を取り巻く空気との相互作用により、各種の汚染物質の光散乱パラメータによって生じます。

スクリーンやモニターの効果：指紋やホコリなどは、モニターがオンになっているときにはほとんど見えませんが、オフになっていると非常に目立ちます。

Kerkytheaでは光散乱パラメータが素材全体に適用され、個別の層には適用されないため、これらの効果はマテリアルエディタで拡散、反射、透過、または半透明チャンネルを使用してのみシミュレートできます。

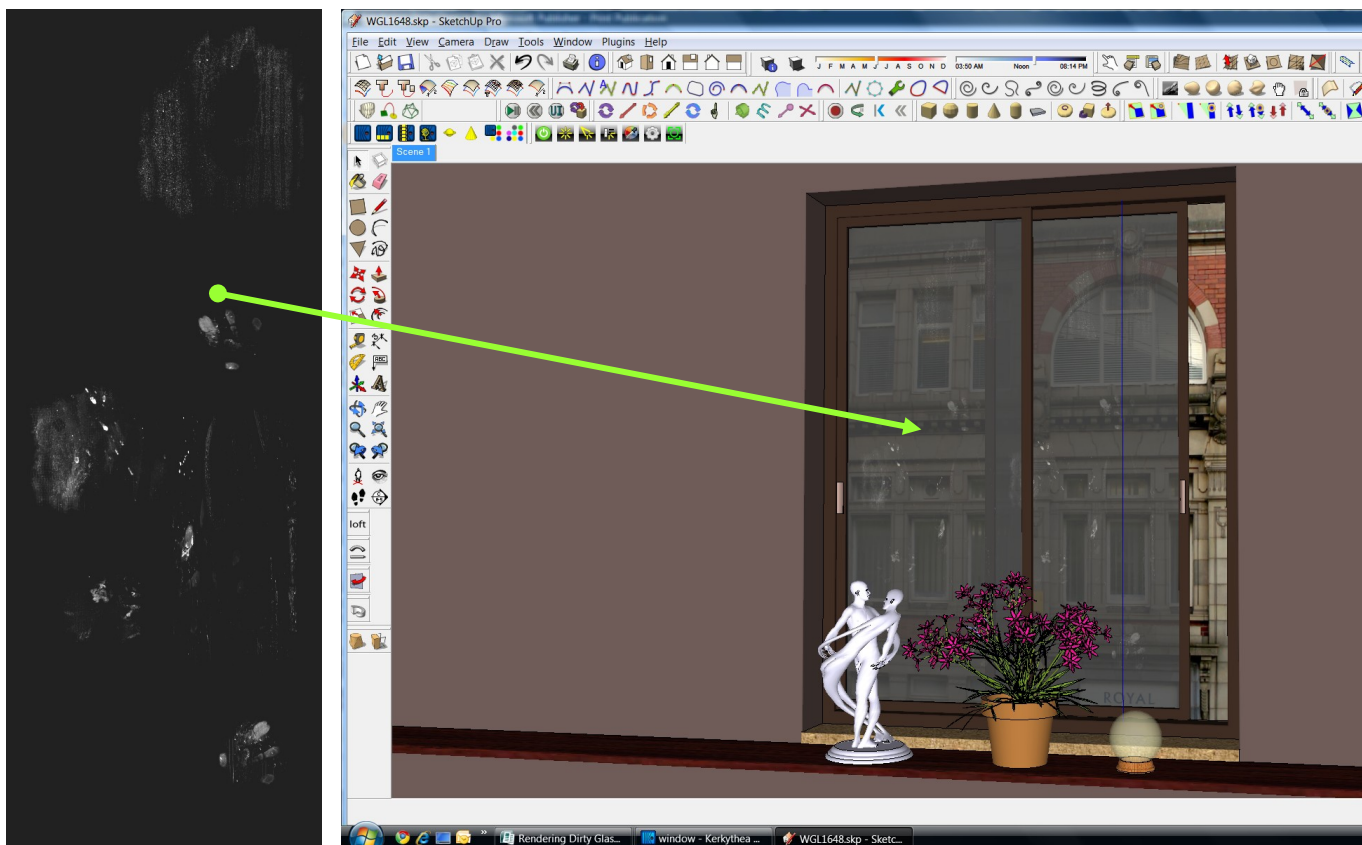
以下に示すマテリアルの作成方法は、物理的に100%正確ではないかもしれませんが、私の経験上、信頼性のある画像を生成できます。この方法が唯一の正解だとは主張しませんが、試行錯誤を経て学んだ結果です。

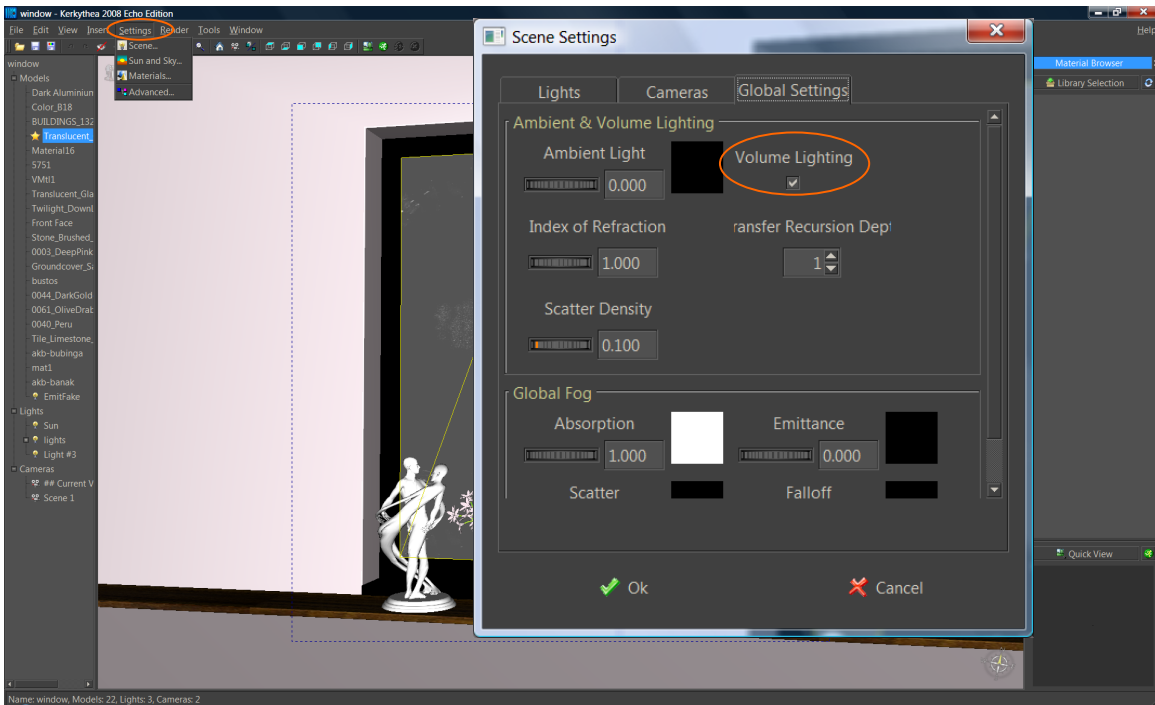
「万能のマテリアル」を作成することはほぼ不可能です。なぜなら、私たちの作業の各シーンは異なり、レンダリングで伝えたい雰囲気によっても変わるからです。そこで、特定のニーズに合った素材をゼロから作成する方法に焦点を当てます。

最初のステップは、Photoshop、Corel Paint、Gimpなどの画像編集ソフトを使用して、作業対象の表面に関連するビットマップを作成することです。これにより、異なるテクスチャをレイヤーごとに黒い背景上にブレンドできるようにします。このビットマップは、Kerkytheaでのメインの拡散レイヤーとなります。

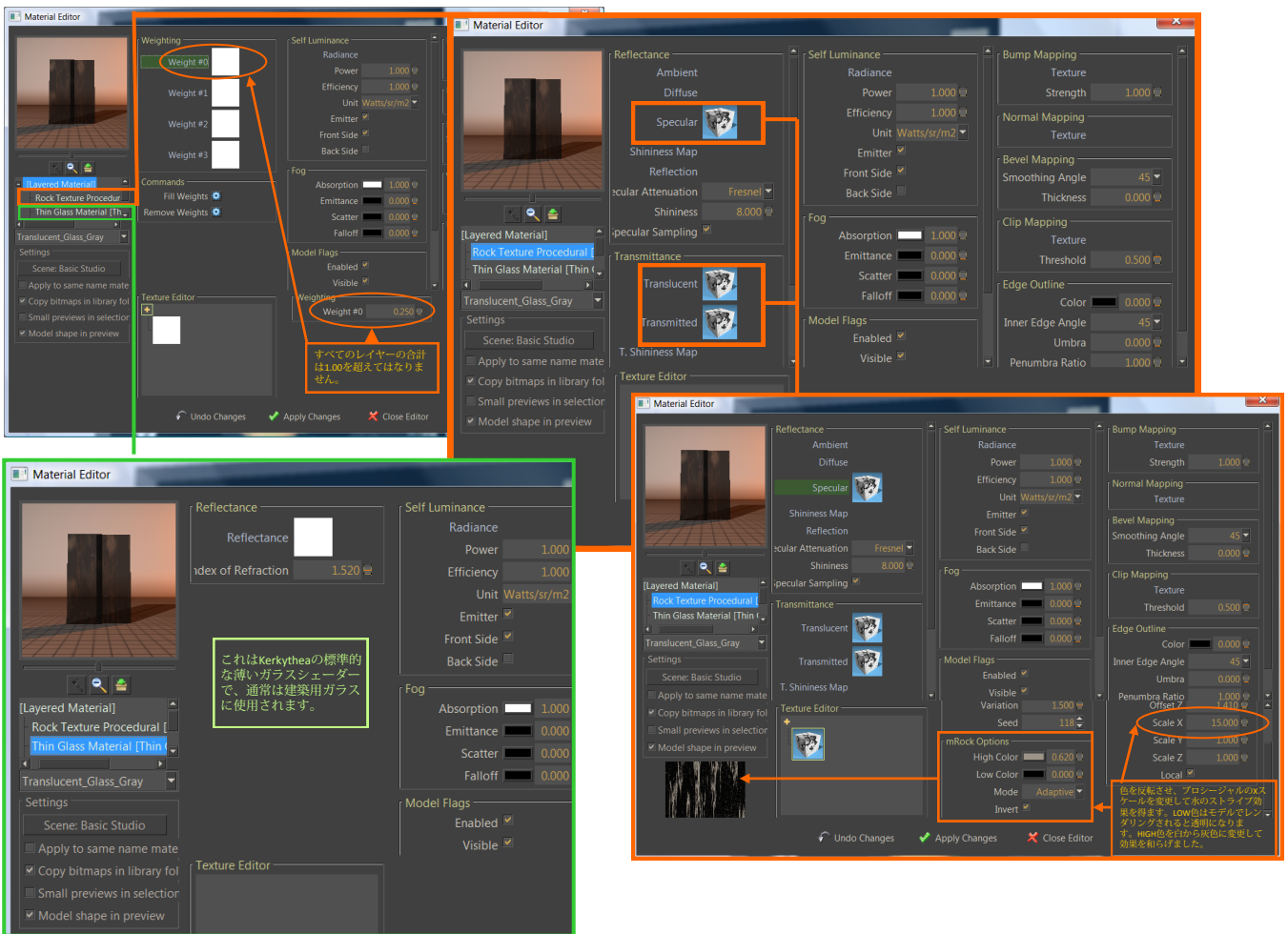
1ページ目のレンダリングでは、窓のガラスの形状と同じ形状の画像を作成し、SketchUpでテクスチャを直接適用およびスケリングしました。そのため、KerkytheaにエクスポートしたときにUV座標を保持することができます。

モデルをエクスポートしてKerkytheaで開いた後、透過と半透明を使用するため、シーンでボリュームライティングを有効にする必要があります。これを行うには、[設定]>[シーン]>[グローバル設定]に進み、ボリュームライティングにチェックを入れます。

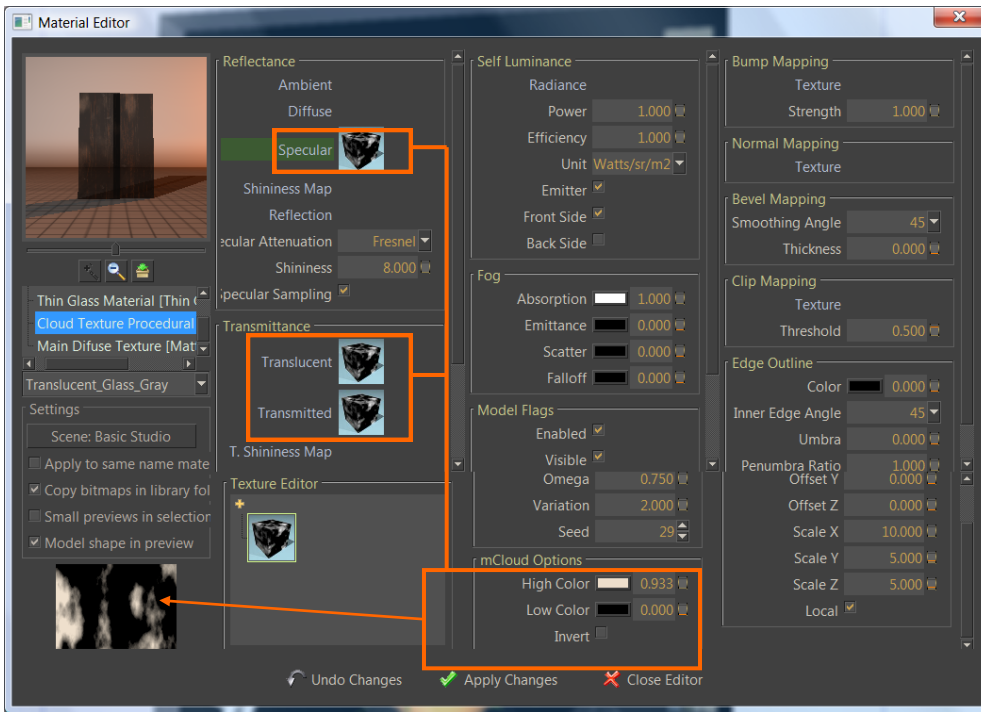




汚れたガラスの材料は、薄いガラスとテクスチャを重ね、岩のプロシージャルと雲のプロシージャルを組み合わせることで作成します。これにより、最後の2つからより面白い効果を得ることができます。プロシージャルはUVマッピングに依存しないためです。



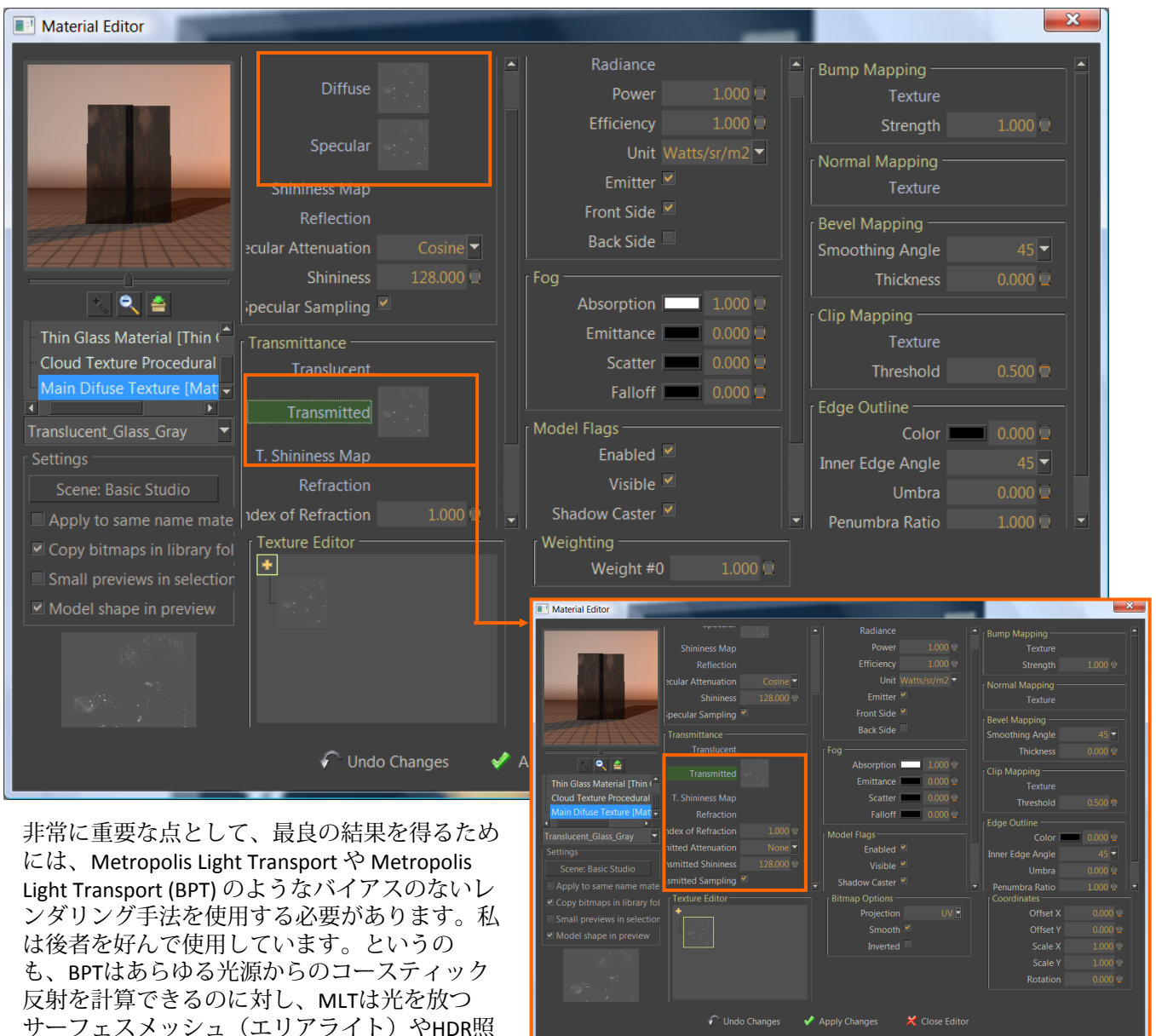
材料は受け取る光よりも多くの光を反射できないため、材料のすべてのコンポーネントレイヤーの重みの合計は1.00を超えてはなりません。ただし、各レイヤーの異なる値を実験することで、各要素から得たい特徴を強調したり、強調しなかったりすることができます。



雲のプロシーダルのを、岩のプロシーダルのと同じ技術を使用して作成します。唯一の違いは、高色と低色のコントラストが高く、テクスチャのスケールが異なることです。これらの要素はすべて、個々のニーズに合わせて調整でき、より強い効果や微妙な効果を得ることができます。

最後に、画像編集ソフトウェアで作成したテクスチャを含む最終レイヤーを組み立てます。

拡散、反射、透過チャンネルに同じテクスチャを使用し、透過サンプリングをオンにします。



非常に重要な点として、最良の結果を得るためには、Metropolis Light Transport や Metropolis Light Transport (BPT) のようなバイアスのないレンダリング手法を使用する必要があります。私は後者を好んで使用しています。というのも、BPTはあらゆる光源からのコースティック反射を計算できるのに対し、MLTは光を放つサーフェスメッシュ（エアライト）やHDR照明によってのみコースティクスを計算するためです。

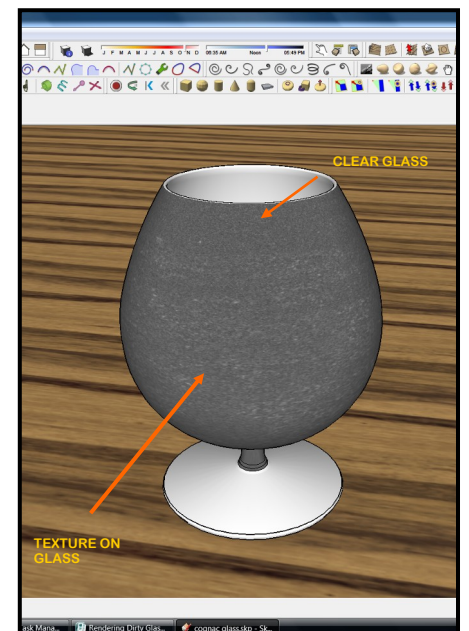
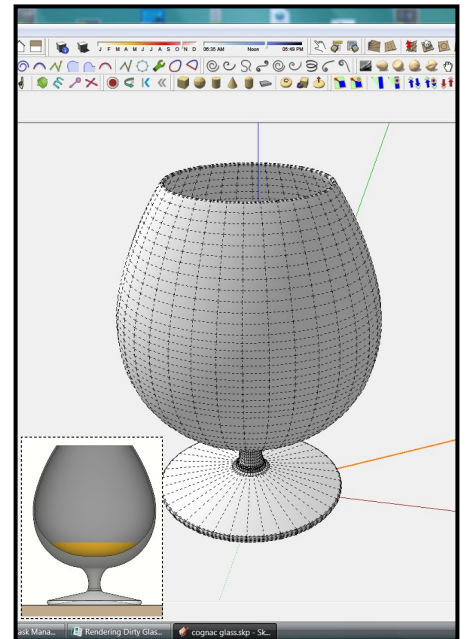


これらは、部屋の内側と外側からの眺めを示す最終画像です。

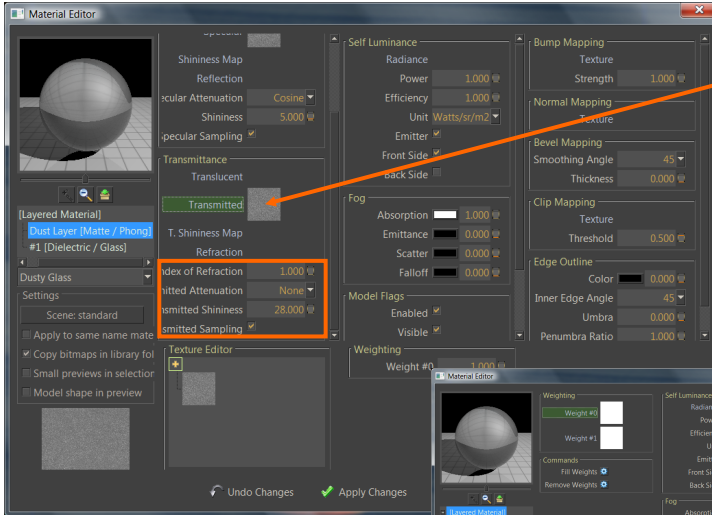
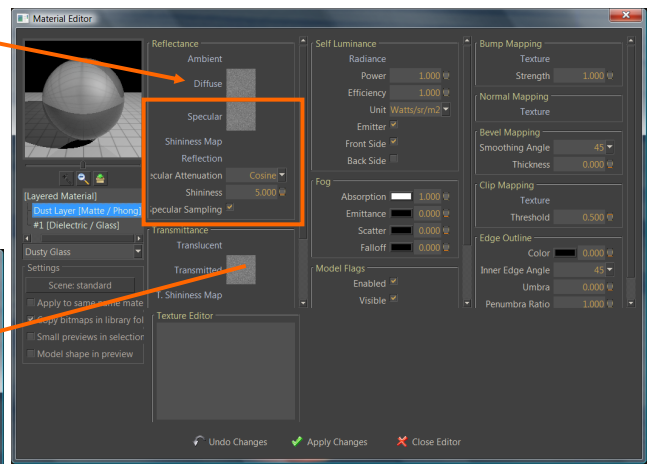
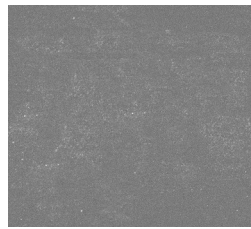
コニャックグラスの研究

もう1つ興味深いシナリオは、コニャックグラスのような透明な表面に薄いホコリの層が与えるコースティック反射への影響です。この演習では、ベースとして薄いガラス素材を使用する代わりに、誘電体ガラスを使用する必要があります。

まず、実物同様に厚みを持たせてコニャックグラスをモデリングします。その後、ホコリの層を構成するビットマップテクスチャを外側の面にのみ適用します (KTリポジトリにあるPatrick Nieborgによる、ガラス内部の液体のモデリング方法を解説した優れたチュートリアルがあります)。



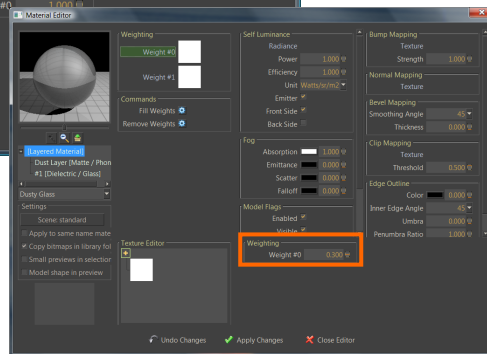
ガラス上のホコリを作成するために、マテリアルのマット/フオンレイヤーの拡散、反射、および透過チャンネルにホコリのテクスチャを適用します。**反射サンプリングを有効にし**、光沢の値を低く設定する必要があります。



透過チャンネルには28のような低い値を設定し、**透過サンプリングを有効にします**。

これで、ガラス上のホコリの効果を増減させるために、マテリアル内の各レイヤーの相対的な重みを調整できます。

下の画像は、ガラスやコニャックからのコースティック反射が、マテリアル内の各コンポーネントに割り当てた値によってどのように減衰するかを示しています。



ガラスのステムからの完全なコースティック反射を含む



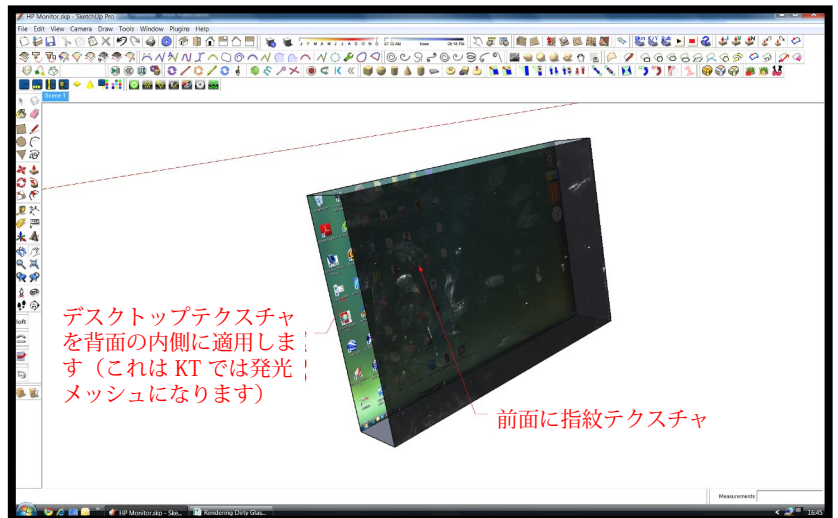
ほこりの層により、コースティック反射は減衰する

スクリーンモニター効果

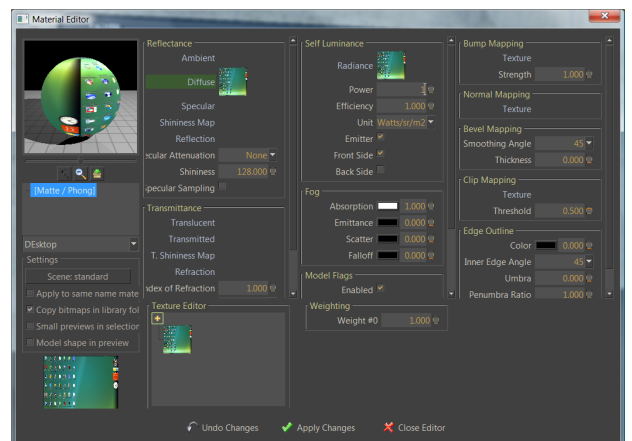
ホコリ、汚れ、指紋のような脂質は、モニター画面などの表面で光との相互作用に応じて異なる振る舞いをします。

下の画像は、モニターがオンのときに画面上の汚染物質が色あせた暗い斑点のように見え、カメラの角度による入射角に応じて減衰する様子を示しています。これは現実でも起こることです。

ここでのポイントは、モニター画面に厚さ（例えば1mm）を持たせてモデル化することです。そして、ガラスの前面には汚染物質のテクスチャを適用し、画面の背面内部にはデスクトップのテクスチャを適用します。モニターがオンであるこ



とをシミュレートするために、このテクスチャをKTでライトエミッターに変換します。

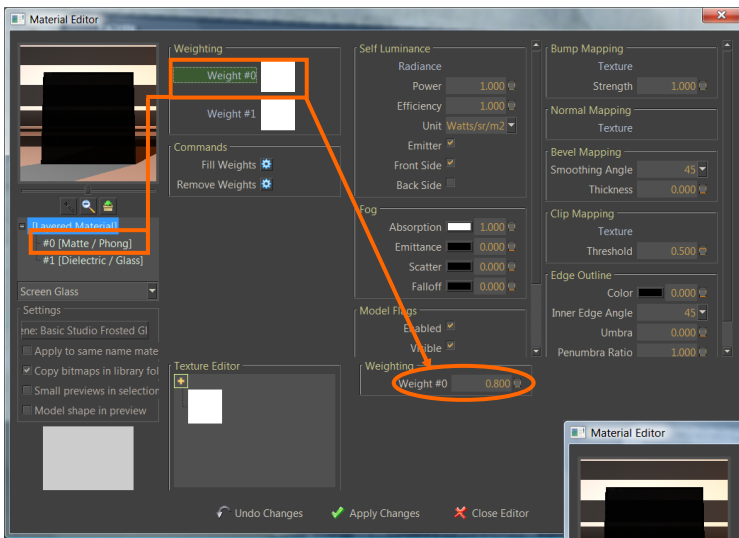


モニターの指紋を作成するために、誘電体ガラスの材料の上に拡散テクスチャを重ねたレイヤード材料を使用しますが、一工夫加えます。

マット/フォン材料は全体の80%を占め、誘電体ガラスが残りの20%を占めます。

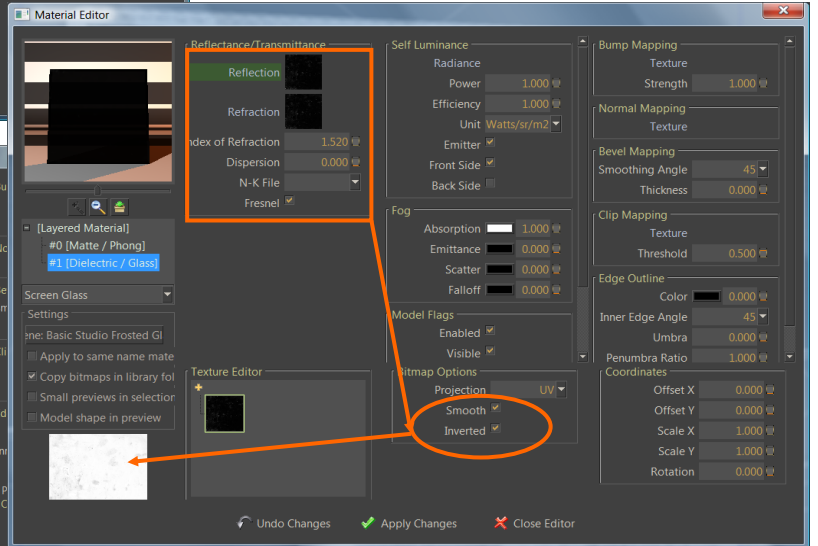
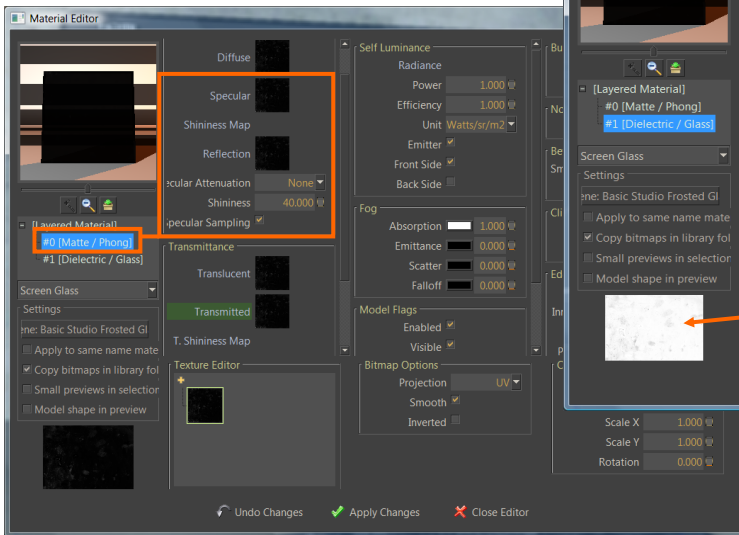


指紋のテクスチャを、材料の拡散、反射、半透明、そして透過のチャンネルに適用します。光沢を約40に減少させ、反射サンプリングを有効にし、透過の設定で調整します。

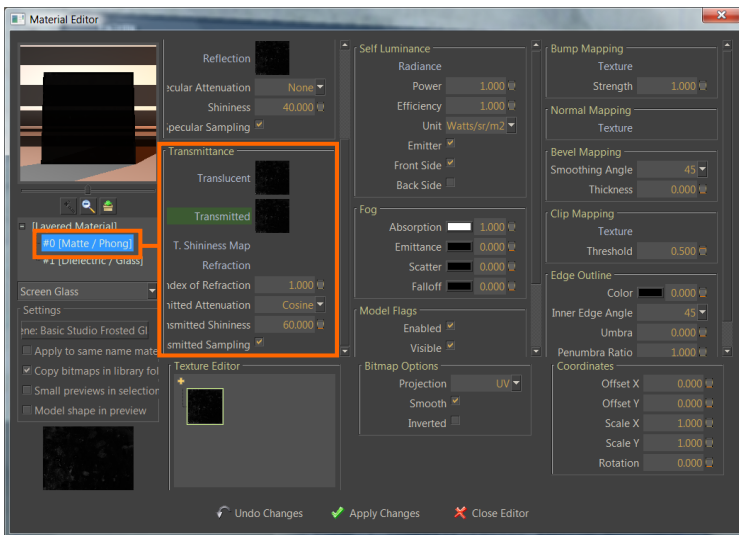


セクションで、透過サンプリングを有効にし、透過の光沢を約60に減少させます。

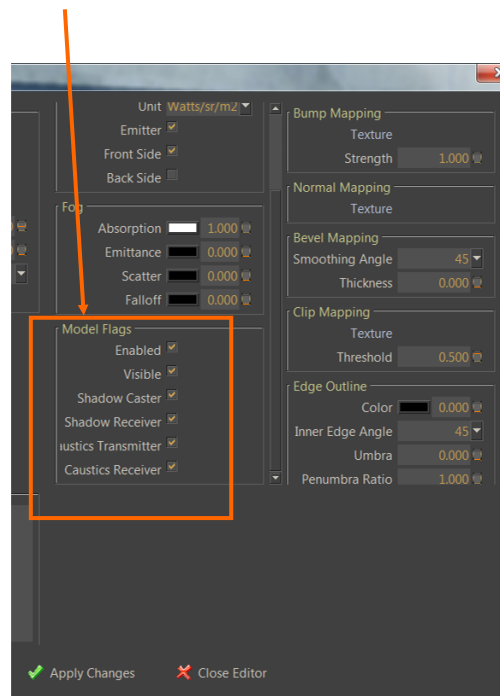
モニター画面は、指紋や汚れで覆われていない部分について、反射的かつ屈折的である必要があります。そのため、同じテクスチャを反転させて使用します。これは、拡散テクスチャの黒い部分がすべて、誘電体ガラスのように振る舞うことを意味します。一方、汚れや指紋の外観は、マテリアルのマットレイヤーで使用した設定によって制御されます。

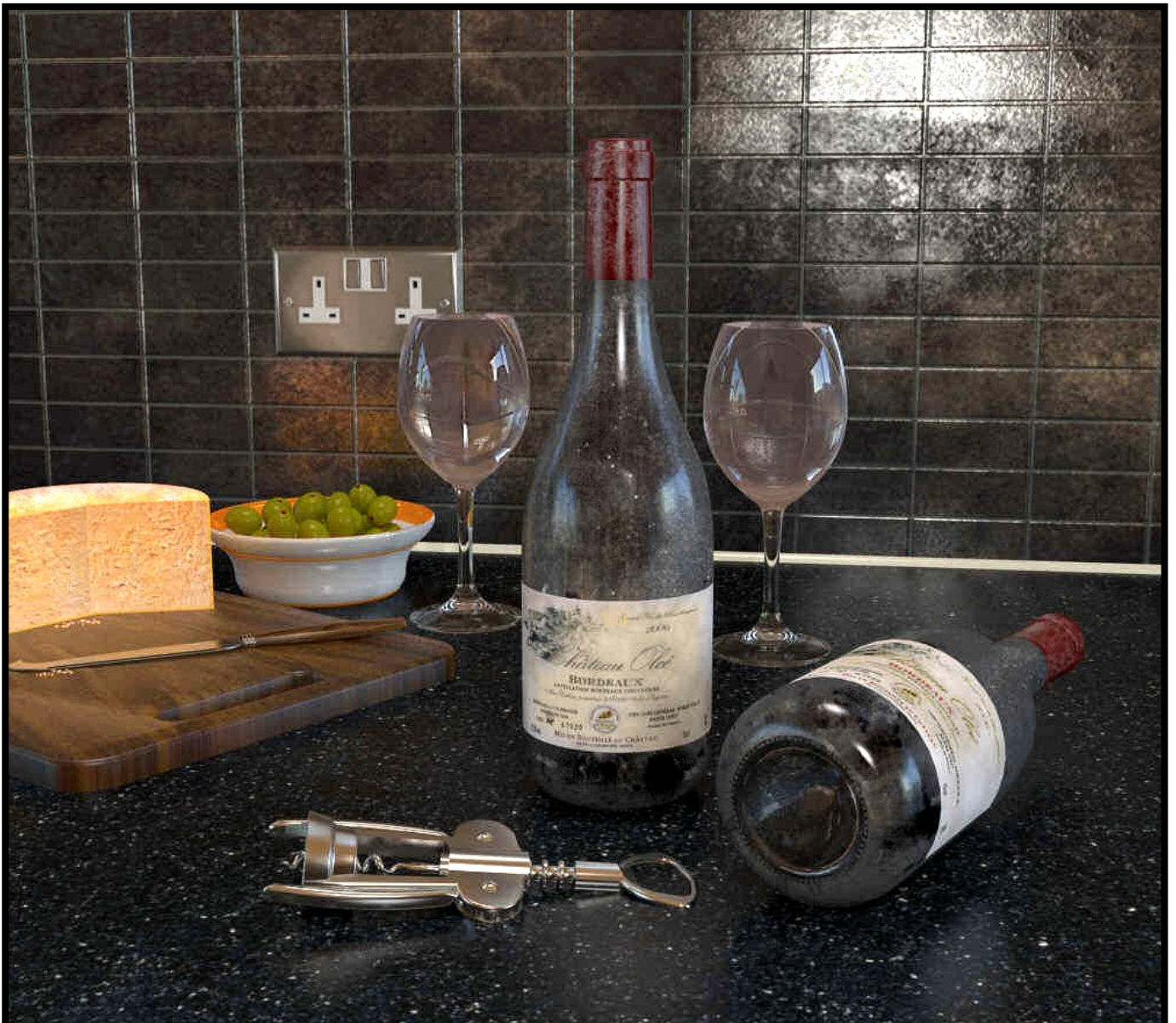


このチュートリアルを締めくくるために、Twilightで提供されている屋外灯の1つを使って、同じ技法で汚れたガラスの昼夜レンダリングを作成しました。1ページ目の窓で説明した技法と同じです。



ここでは、直接日光によって照らされたときのマテリアルの挙動と、夜間に光がガラスの後ろから来るとき、および周囲への影響を見ることができます。汚れの層から投影される影を排除したい場合は、マテリアルエディタのモデルフラグ設定でこの機能をオフにしてください。





このチュートリアルで扱う理論的な概念は、コロンビア大学のJinwei Gu、Ravi Ramamoorthi、Peter Belhumeur、Shree Nayarによって発表された研究「Dirty Glass: Rendering Contamination on Transparent Surfaces」（2007年）に基づいています。

また、このソフトウェアを提供してくれたKTチームの全メンバーと、長年にわたりKTフォーラムでその豊富な経験やクリエイティブな作品を共有してくれたKTコミュニティのメンバーに特別な感謝を捧げます。■

“注目に値するアイデアを生み出せる人は、他者から多くを学んだ人です。”. 松下幸之助
パナソニックの創業者