
POV-Ray

1. 3次元のCGの作成

3次元のCG(Computer Graphics)を体験してみましょう。図1は、3次元のCGを生成するための一般的な手順を示したものです。このような手順にしたがってCGを生成することをレンダリングといいます。POV-Ray(ポブレイ)はこれらの一連の処理を行うことができるソフトウェアです。CGの理論等については、関連する専門科目で学んで下さい。

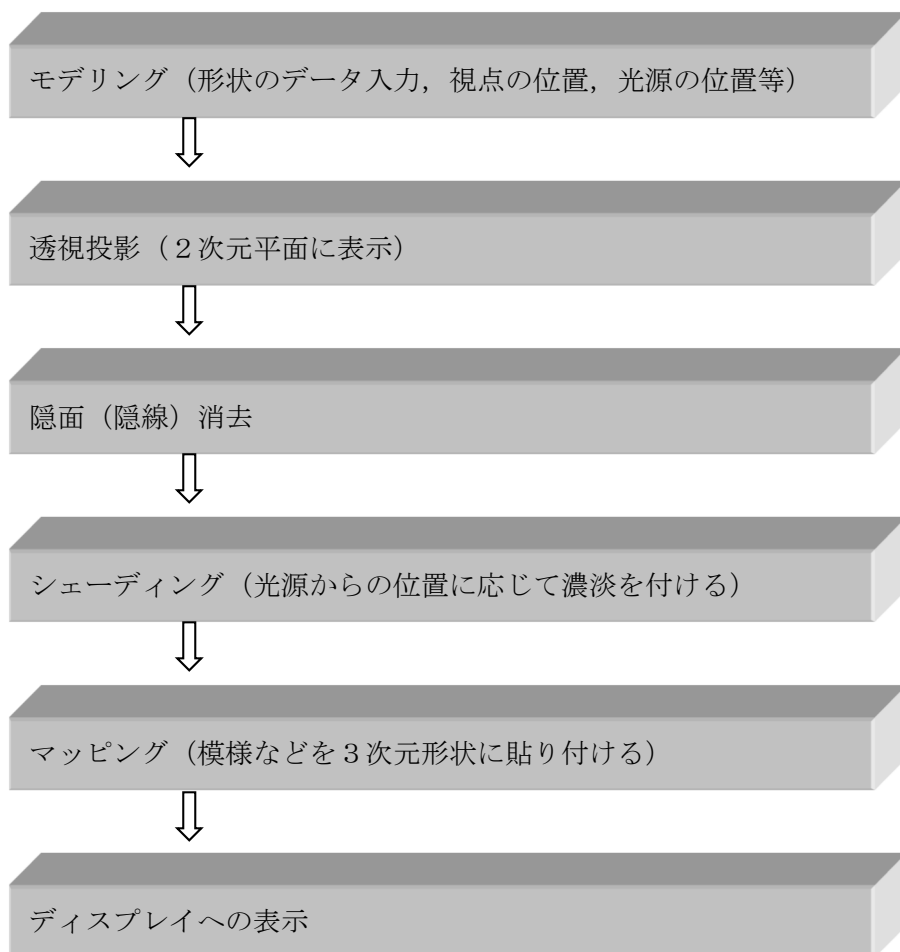


図1 3次元CGの生成手順

[参考資料]

- [1] ビジュアル情報処理 -CG・画像処理入門-, CG-ARTS 協会
- [2] コンピュータグラフィックス, CG-ARTS 協会
- [3] POV-Ray で学ぶ実習コンピュータグラフィックス, アスキー出版局
- [4] <http://www.povray.org/>

2. POV-Ray の座標系

(1) 座標系 (左手系)

3次元のCGを生成するためには、表示する物体の形状をデータで入力する必要があります。データは3次元の直交座標系で指定します。POV-Rayの座標系は、左手系になっています。ちょうど、左手の親指がX軸、人差し指がY軸、中指がZ軸に対応していることから、このように呼ばれています。この様子を図2に示します。

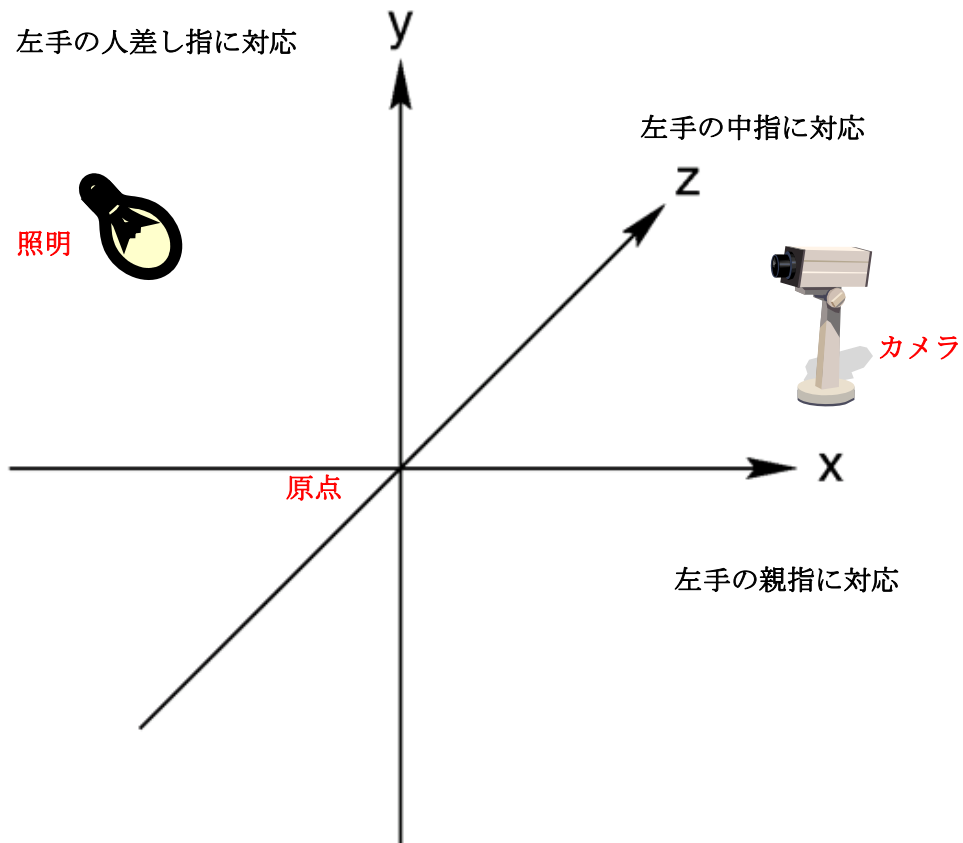


図 2 POV-Ray の座標系 (左手系)

(2) 軸周りの正の方向

軸周りの正の方向は、左手の親指を軸の正の方向に向けて握った場合、4本の指の方向になります。この様子を図3に示します。

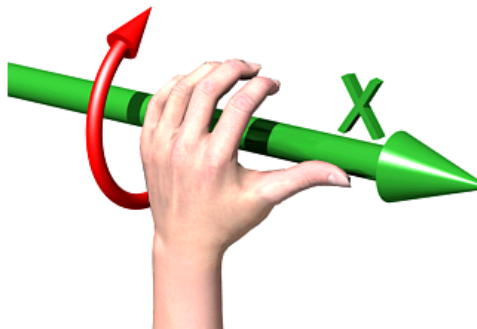


図 3 軸周りの正の方向

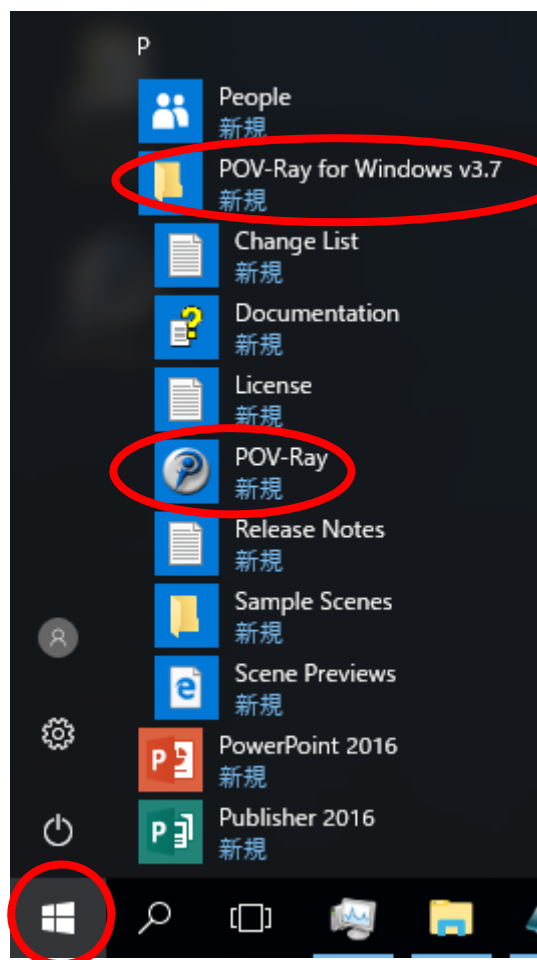
3. POV-Ray の起動

それでは、POV-Ray を起動してみましょう。POV-Ray をまだインストールしていない人は、速やかに行ってください。

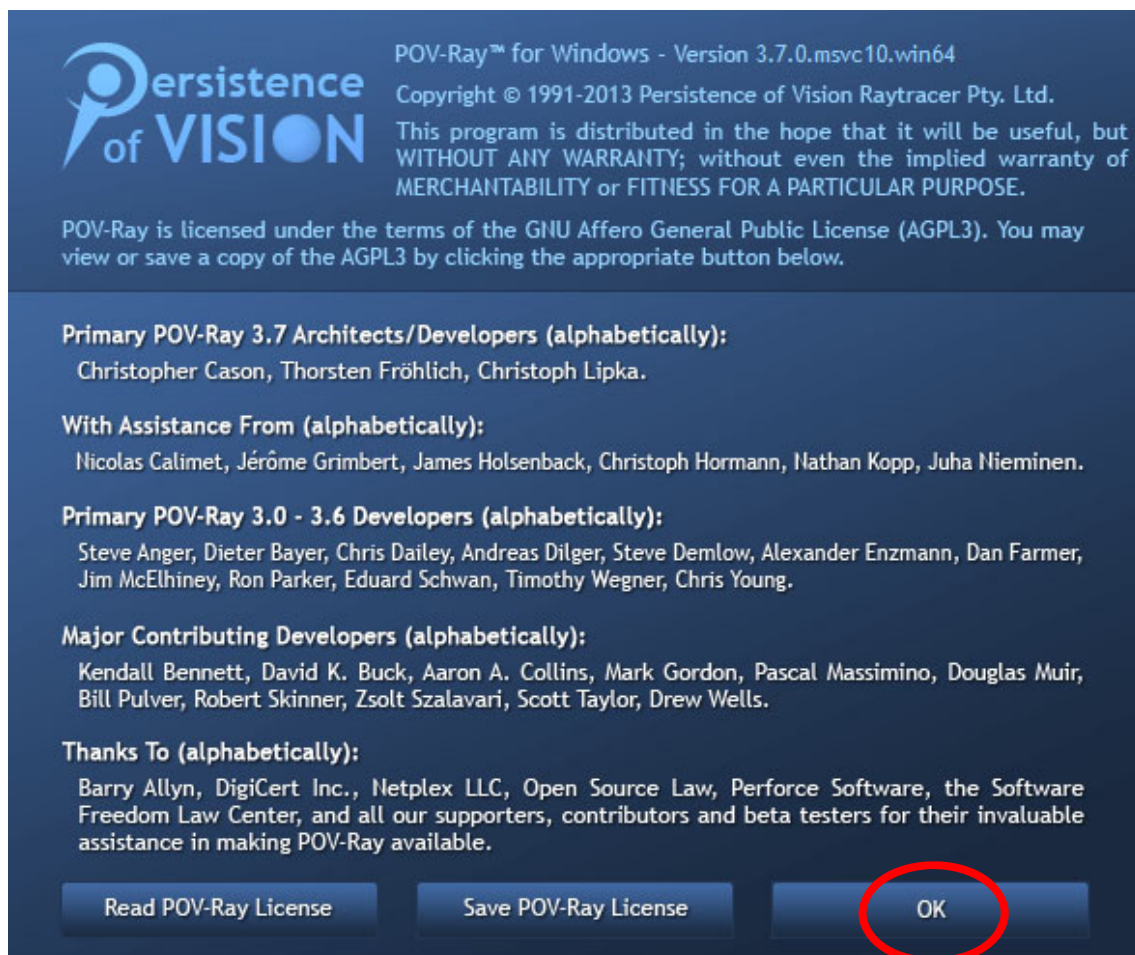
- (1) デスクトップ上の「POV-Ray v3.7」のアイコンをダブルクリックします。



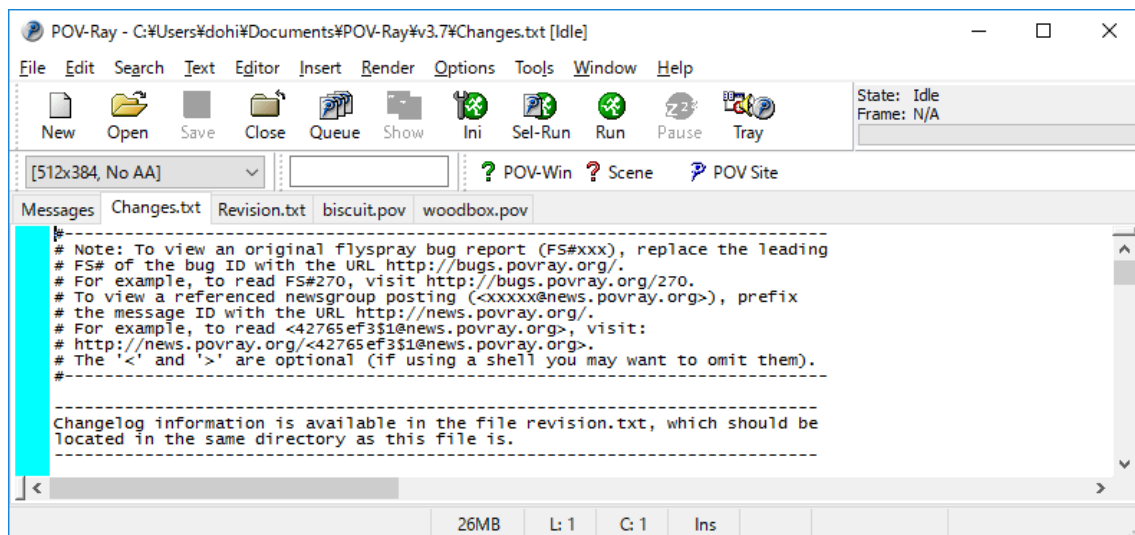
- (2) デスクトップ上に見当たらない場合は、「スタート」, 「POV-Ray for Windows v3.7」, 「POV-Ray」の順にクリックします。



(3) しばらくすると、以下のウインドウが表示されますので、[OK] をクリックします。

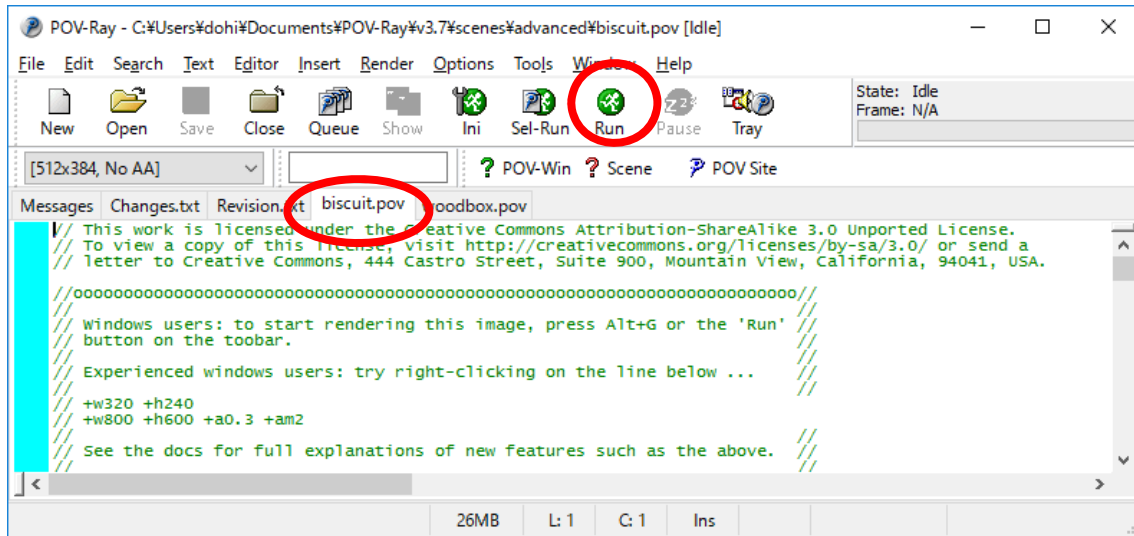


(4) 以下のウインドウが表示され、POV-Ray の起動が完了します。

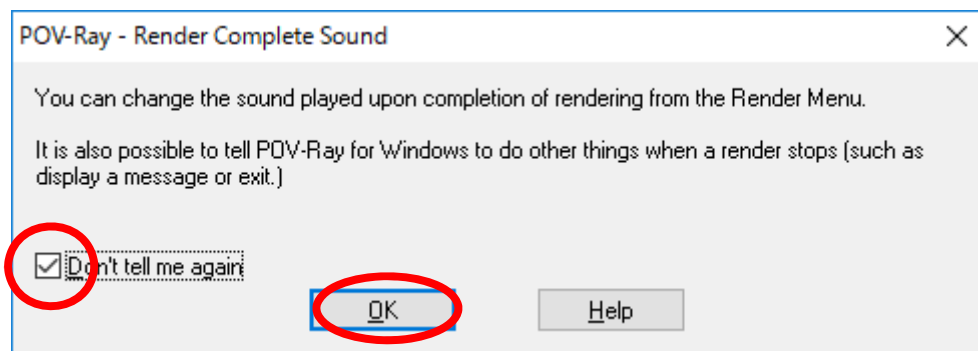


(5) レンダリング

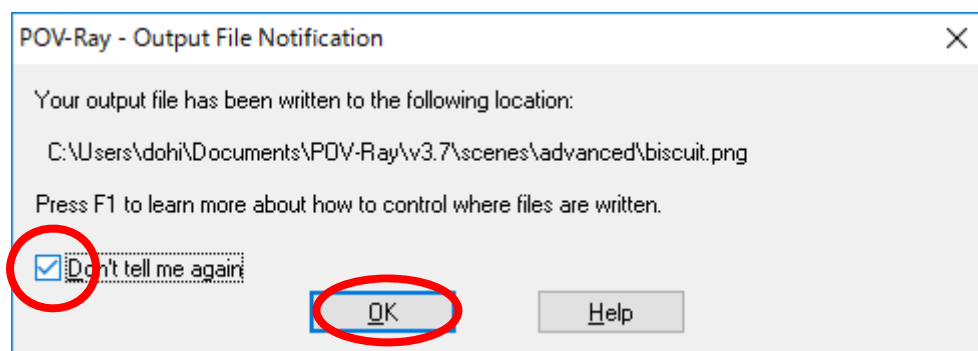
POV-Rayには、あらかじめシーンファイルのサンプルが用意されています。レンダリングが機能することを確認しましょう。「biscuit.pov」タブをクリックした後、「Run」をクリックします。POV-Rayのファイルは、シーンファイルと呼ばれています。



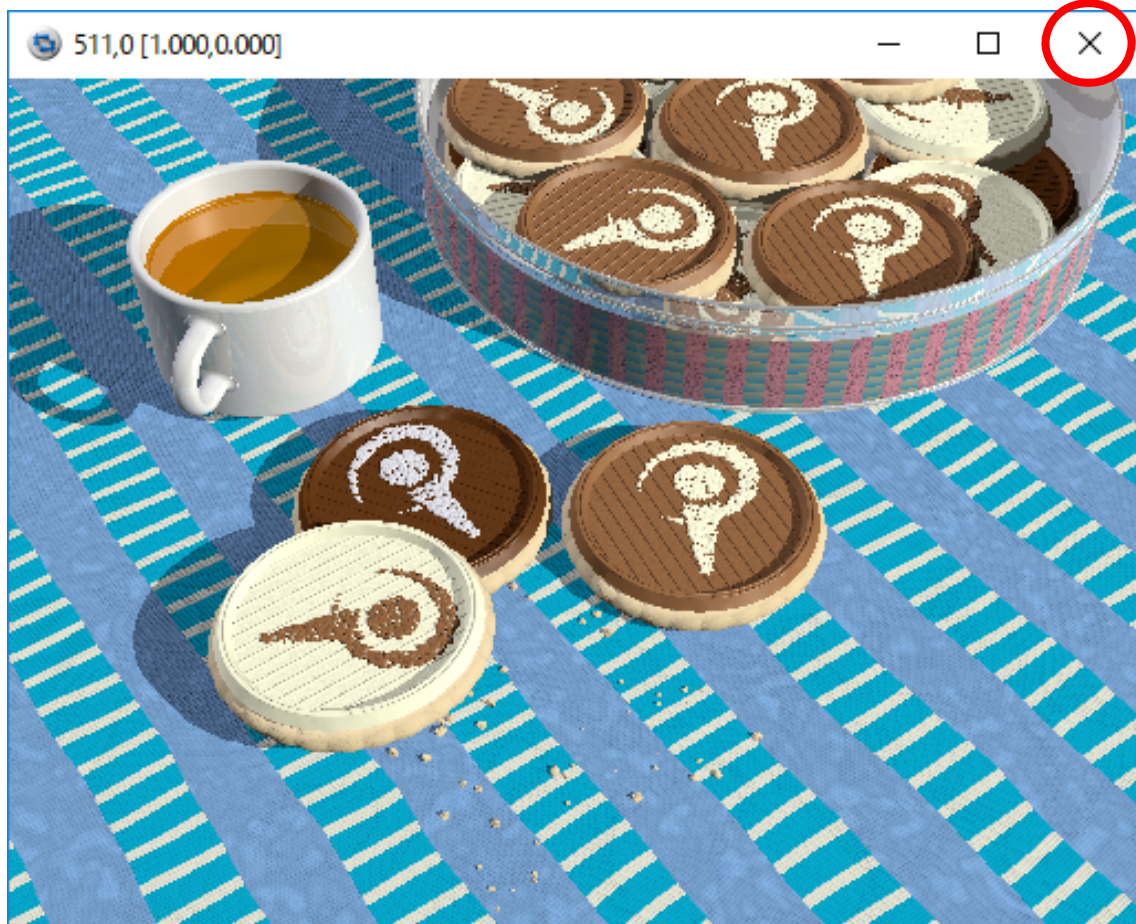
(6) 「Don't tell me again」をチェックした後、「OK」をクリックします。



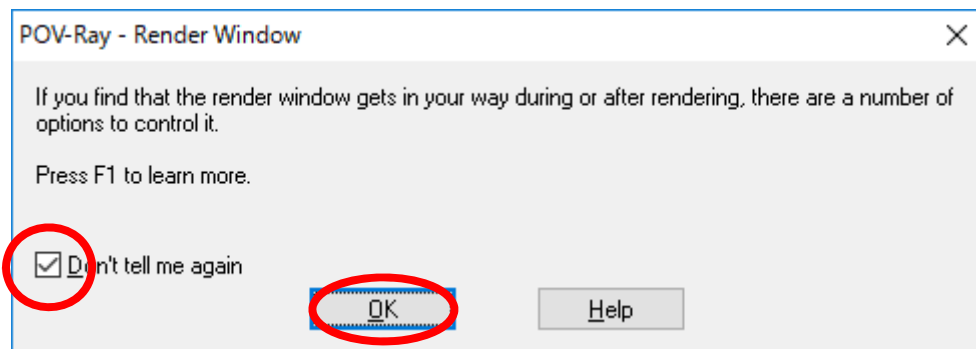
(7) 「Don't tell me again」をチェックした後、「OK」をクリックします。



(8) レンダリングの結果が表示されます。表示内容を確認し、ウインドウを閉じます。



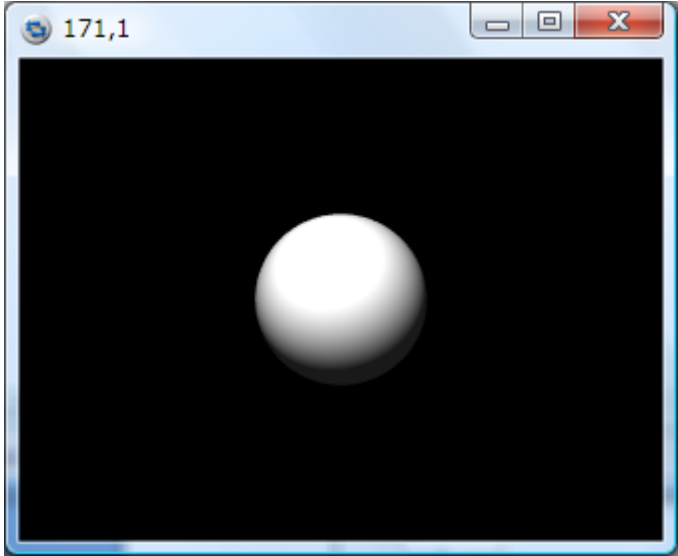
(9) 「Don't tell me again」をチェックした後、「OK」をクリックします。



4. シーンファイルの作成

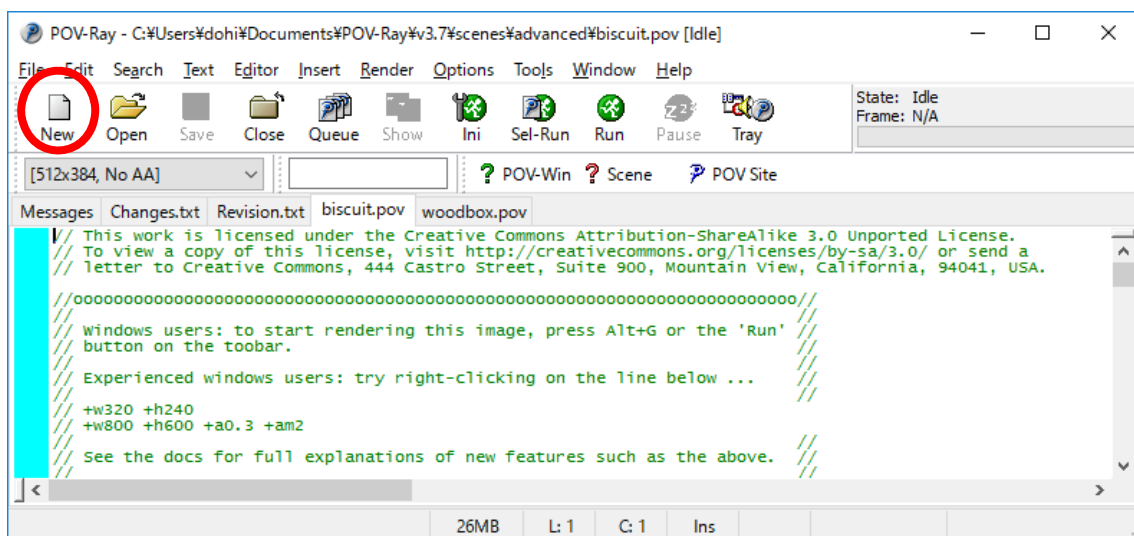
(1) 球の表示

最初の例は、原点に球の中心を置き、半径 1 の白色の球を表示してみましょう。カメラの位置は、(5, 5, -20)とし撮影範囲は 20° とします。この条件を満足するシーンファイルは、以下のように記述できます。

シーンファイル	レンダリング結果
<pre>#include "colors.inc" #include "shapes.inc" #include "textures.inc" camera{ location <5, 5, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } sphere{ <0, 0, 0>, 1 pigment{ color White } }</pre>	

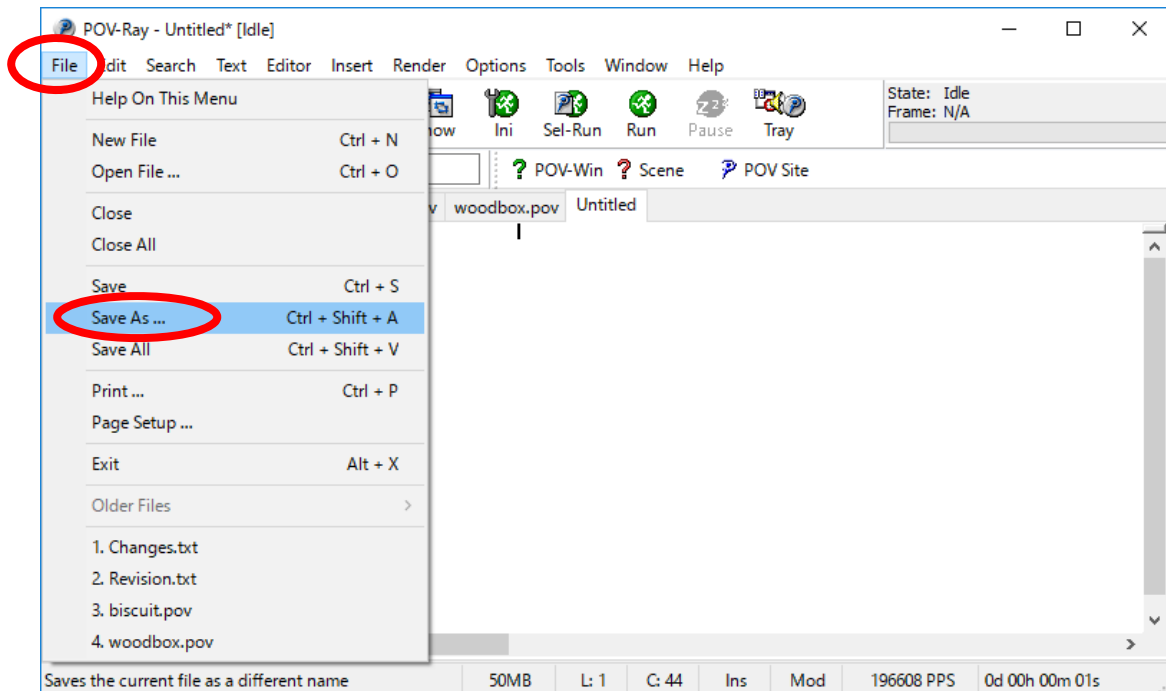
(2) シーンファイルの新規作成

シーンファイルを作成してみましょう。「New」をクリックします。



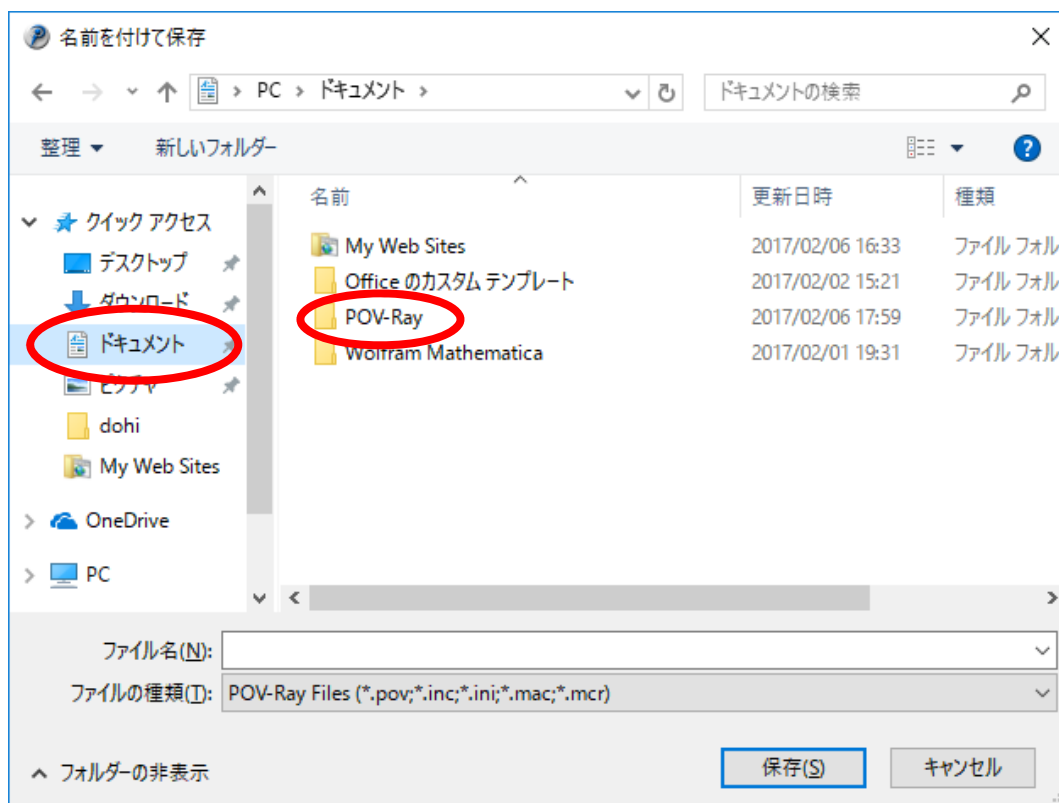
(3) シーンファイルの保存

まだ、何も入力していない状態ですが、「File」「Save As」の順にクリックし保存します。こうすることによって、シーンファイルの入力に応じて予約語等に色が付き、スペルミスなどの誤りを発見し易くなります。



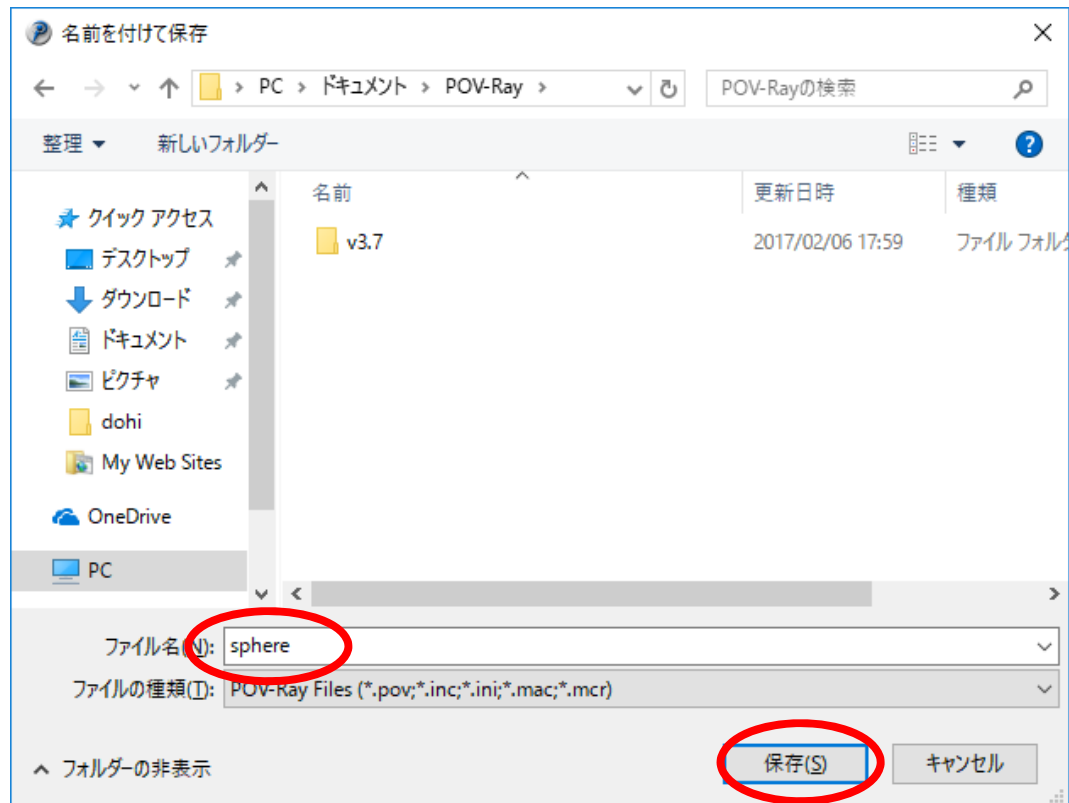
(4) 保存先のフォルダの指定

この例では、ドキュメントフォルダの中の POV-Ray フォルダに保存します。「ドキュメント」をクリックした後、「POV-Ray」をダブルクリックします。

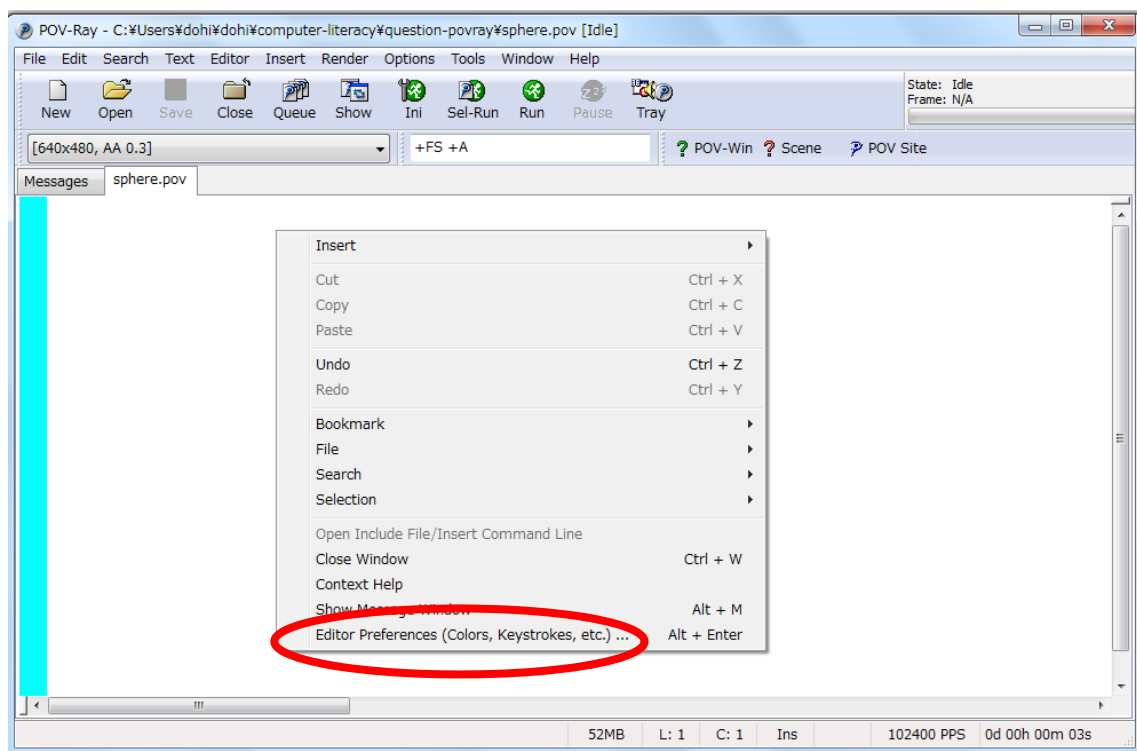


(5) ファイル名の指定

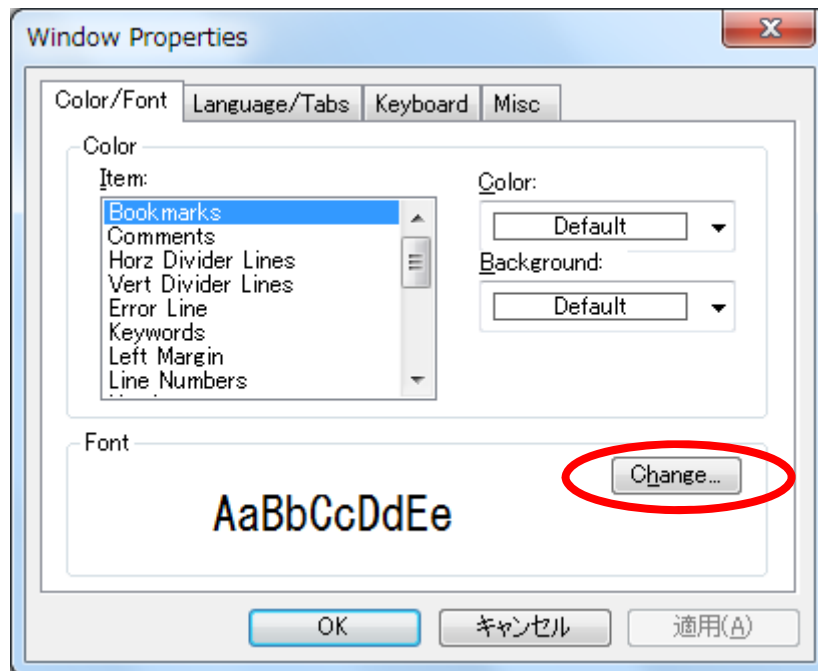
ファイル名は「**sphere**」にしておきます。「**保存**」をクリックします。



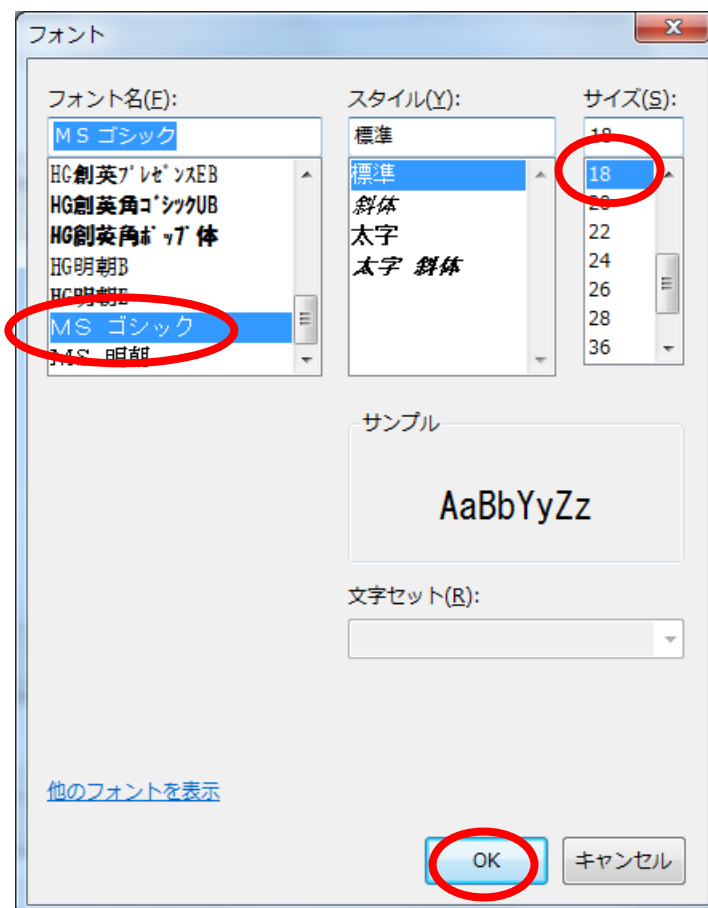
(6) デフォルトのフォントでは日本語が表示できません。文字サイズも含めて変更しておきます。編集画面上を右クリックし、「**Editor Preferences**」をクリックします。



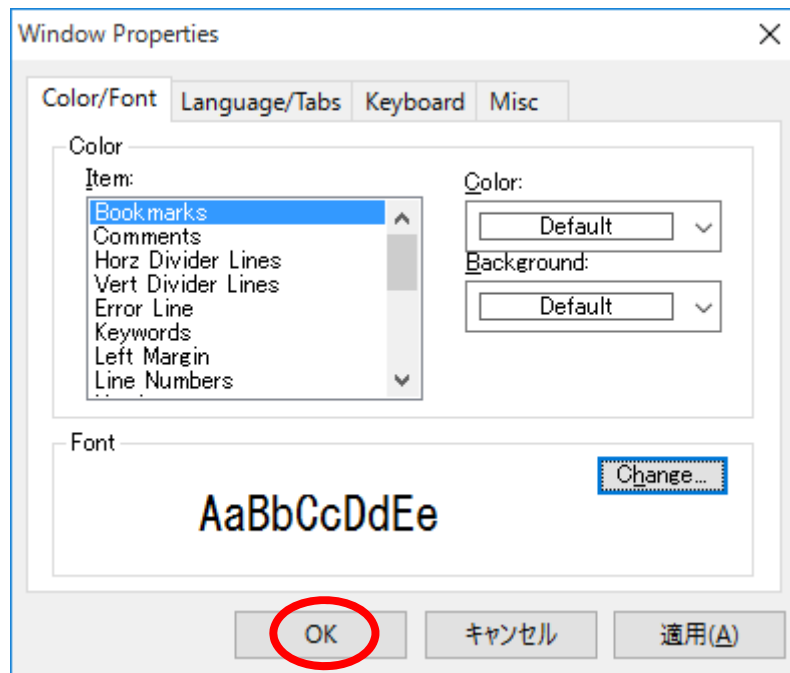
(7) 「Change」をクリックします。



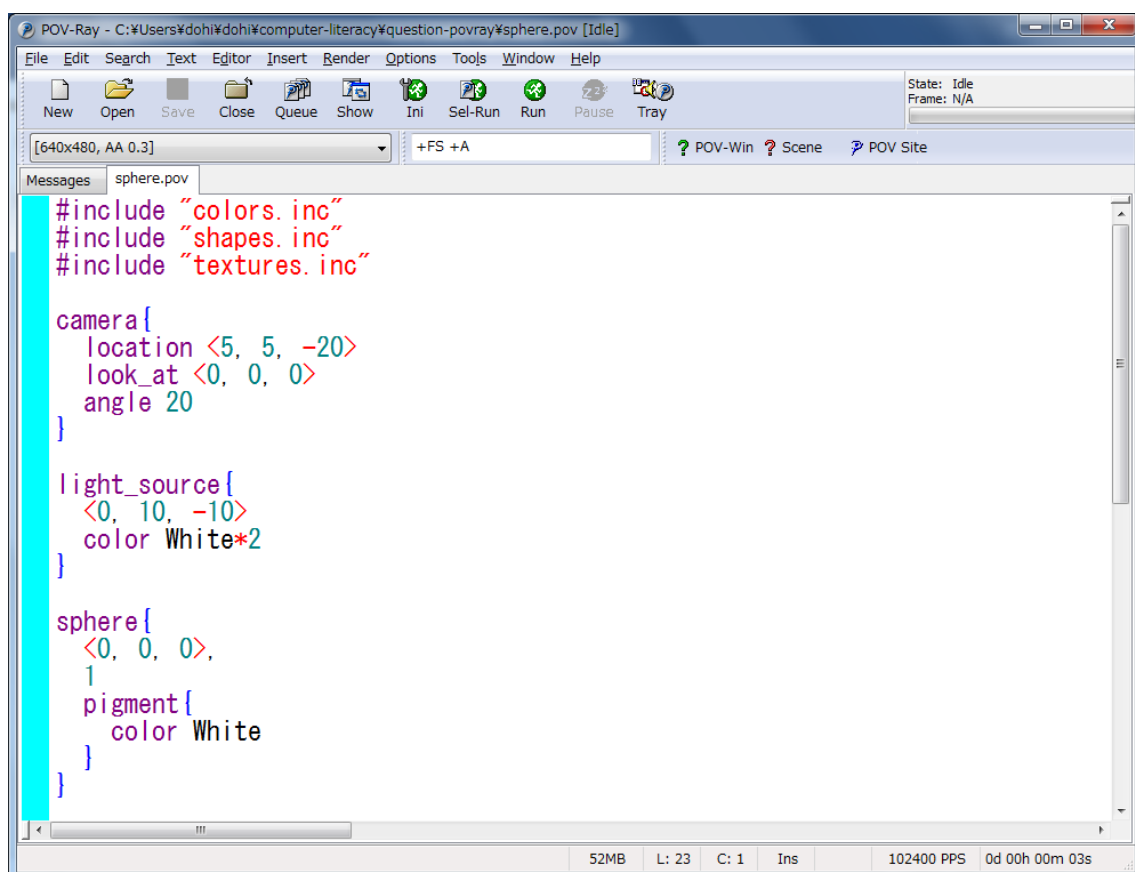
(8) フォントは「MS ゴシック」, サイズは「18」を選びました. 「OK」をクリックします.



(9) 「OK」をクリックします。

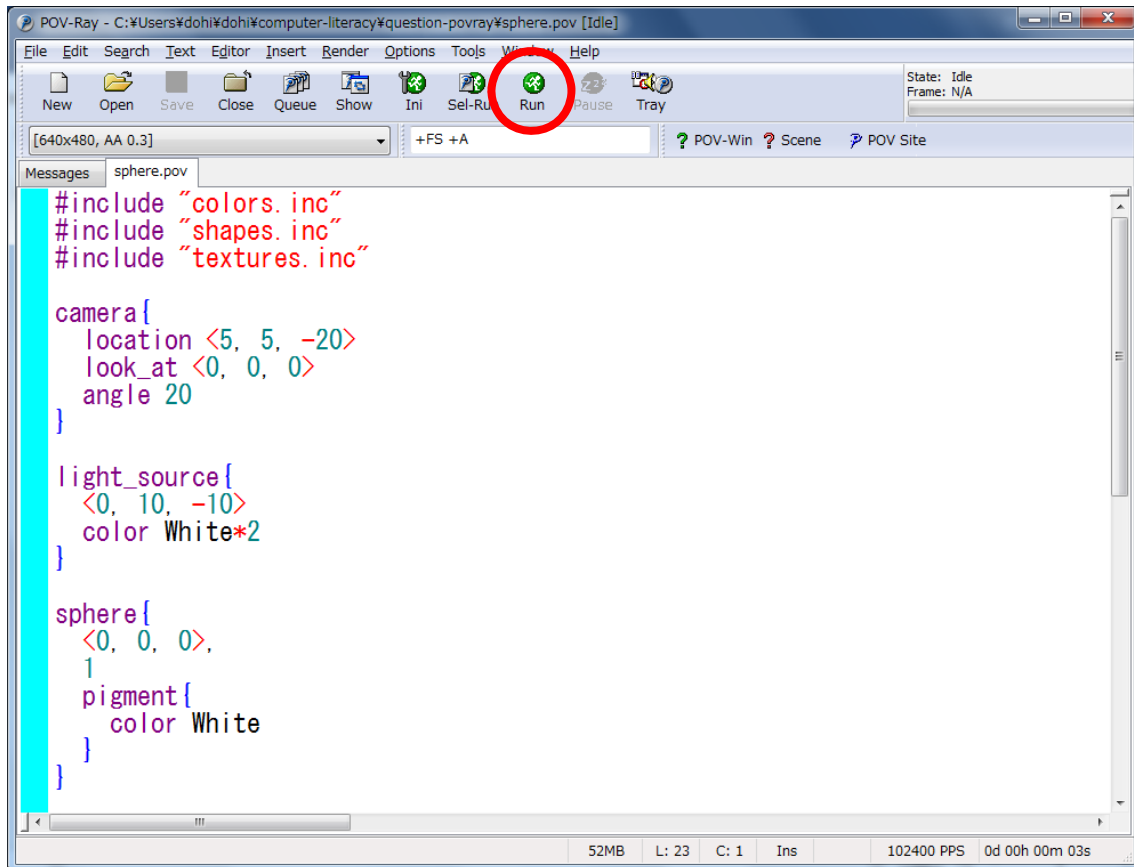


(10) シーンファイルの内容を入力します。

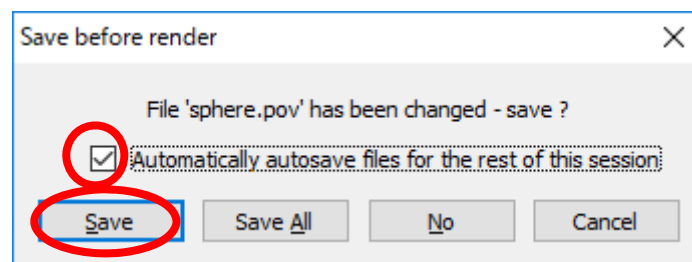


(11) レンダリングの実行

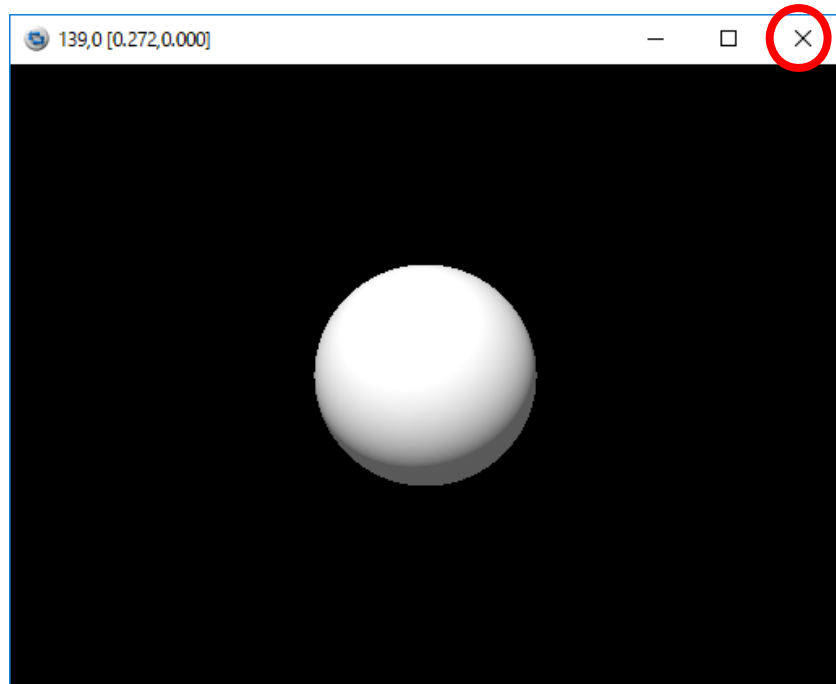
「Run」をクリックすると、レンダリングが始まります。



(12) 以下の表示が出る場合は、「Automatically autosave files for the rest of this session」をチェックし、「Save」をクリックします。



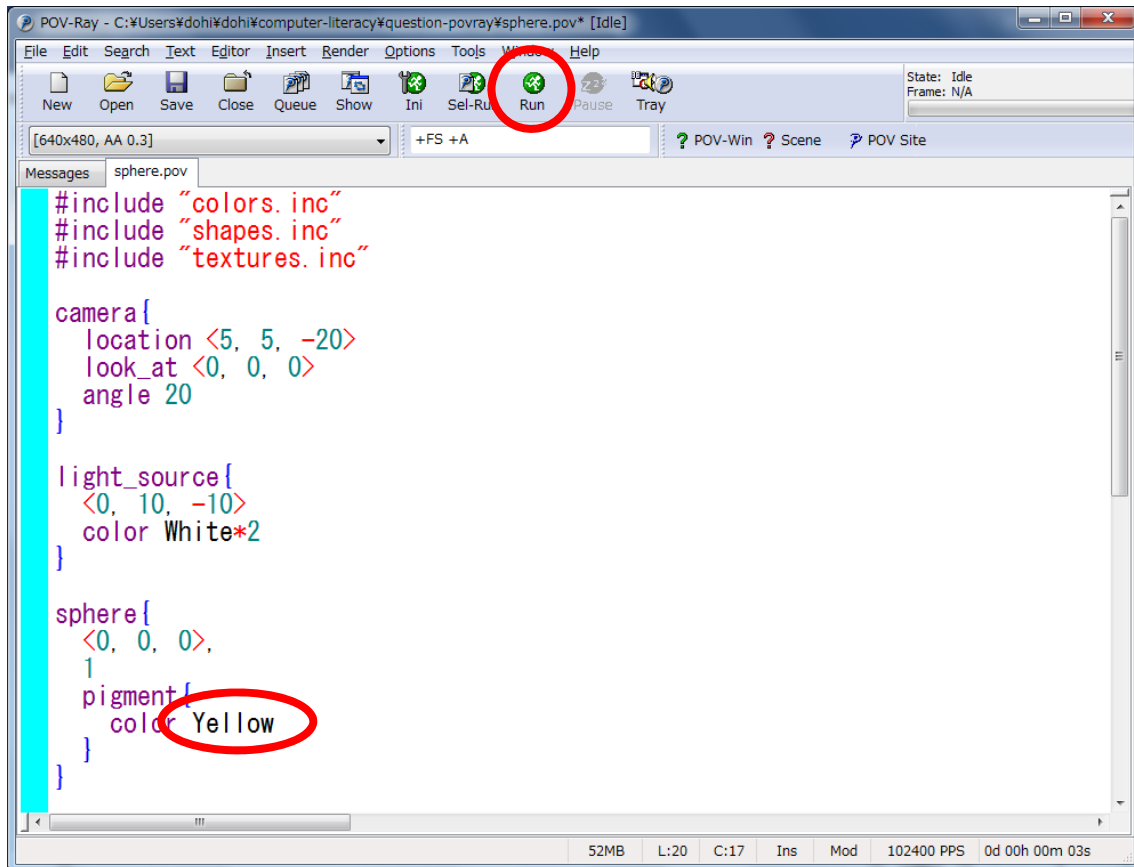
- (13) レンダリングの結果が表示されます。ウインドウの [×] をクリックし、閉じます。



- (14) 球の色を変更してみましょう。色の変更は、`pigment` で指定します。黄色に変更する場合は、"White" を "Yellow" に変更します。

シーンファイル(抜粋)	レンダリング結果
<pre>sphere{ <0, 0, 0>, 1 pigment{ color Yellow } }</pre>	

”Yellow”に変更した後、「Run」をクリックし、レンダリングを開始します。

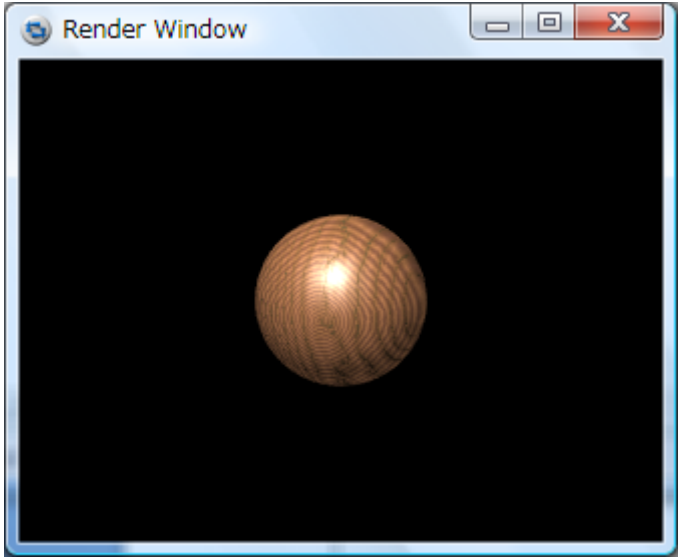


POV-Ray の内部では、色は以下のように定義されています。色々、試してみましょう。

Red Green Blue Yellow Cyan Magenta Clear White Black

5. テクスチャマッピング

球のテクスチャ（物体の模様）を変更する場合は、”DMFWood6”のように入力します。このように物体に模様を貼り付けることを、テクスチャマッピングと言います。

シーンファイル(抜粋)	レンダリング結果
<pre>sphere{ <0, 0, 0>, 1 texture{ DMFWood6 } }</pre>	

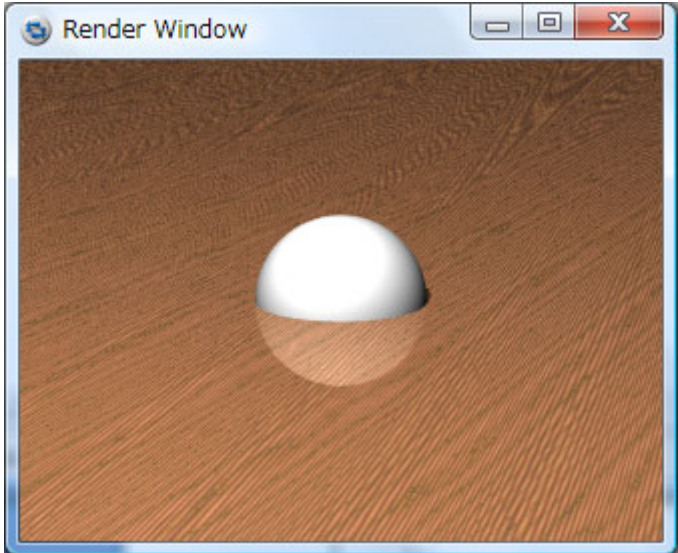
POV-Ray では、以下テクスチャが定義されています。色々、試してみましょう。

DMFWood6	NBglass	NBoldglass	NBwinebottle	NBbeerbottle
Ruby_Glass	Dark_Green_Glass	Yellow_Glass	Orange_Glass	
Vicks_Bottle_Glass	Soft_Silver	New_Penny	Tinny_Brass	
Gold_Nugget	Aluminum	Bright_Bronze	Lightening1	Lightening2
Brushed_Aluminum	Starfield	Shadow_Clouds		

6. 地面の表示

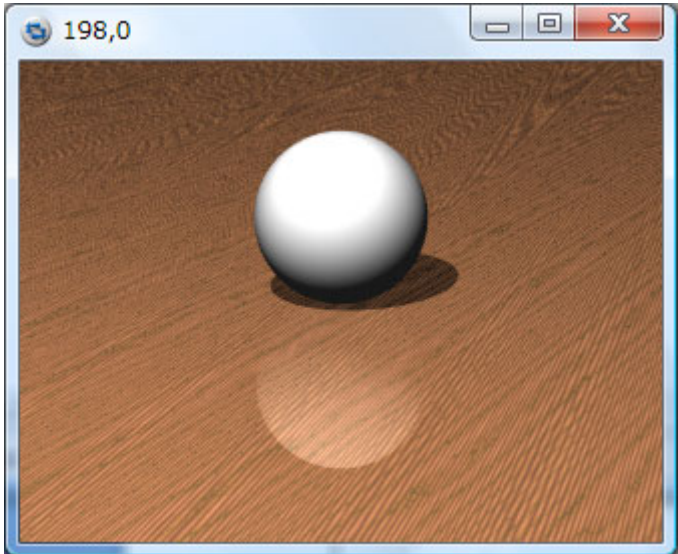
(1) 地面の設定

地面は、object の中で Plane_XZ を使って作成します。地面に対しても、球と同様にテクスチャマッピングを行うことができます。白い球と、地面に DMFWood6 をマッピングした例を示します。

シーンファイル(抜粋)	レンダリング結果
<pre>sphere{ <0, 0, 0>, 1 pigment{ color White } } object{ Plane_XZ texture{ DMFWood6 } }</pre>	 A screenshot of a 'Render Window' showing a 3D scene. A white sphere is positioned on a brown, wood-textured floor. The sphere is partially submerged in the floor, creating a shadow. The window has a blue title bar with the text 'Render Window' and standard window controls.

(2) 地面の設定

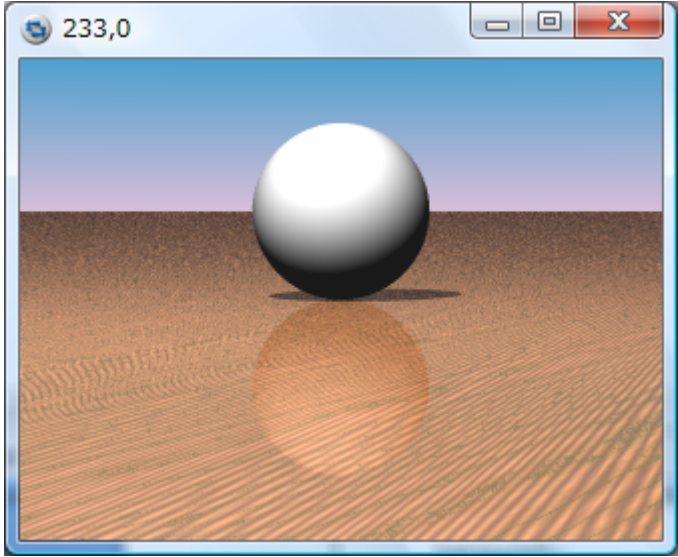
球の中心を $\langle 0, 0, 0 \rangle$ から $\langle 0, 1, 0 \rangle$ に変更すると、球が地面の上に浮き出てきます。

シーンファイル(抜粋)	レンダリング結果
<pre>sphere{ <0, 1, 0>, 1 pigment{ color White } } object{ Plane_XZ texture{ DMFWood6 } }</pre>	 A screenshot of a 'Render Window' showing a 3D scene. A white sphere is floating above a brown, wood-textured floor. The sphere is fully above the floor, creating a shadow. The window has a blue title bar with the text '198,0' and standard window controls.

7. 空の表示

(1) 空の設定

空は, sky_sphere を使って作成します. 空の名前は S_Cloud2 を使いました. なお, カメラの位置を地面近くに移動しないと空が見えません. カメラの位置は $\langle 5, 5, -20 \rangle$ から $\langle 5, 1, -20 \rangle$ へ移動しました. さらに `#include "skies.inc"` を追加してください.

シーンファイル	レンダリング結果
<pre>#include "colors.inc" #include "shapes.inc" #include "textures.inc" #include "skies.inc" camera{ location <5, 1, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } sphere{ <0, 1, 0>, 1 pigment{ color White } } object{ Plane_XZ texture{ DMFWood6 } } sky_sphere{ S_Cloud2 }</pre>	 A screenshot of a POV-Ray rendering window. The window has a title bar with a blue icon, the text "233,0", and standard window controls (minimize, maximize, close). The main area shows a 3D scene: a white sphere sits on a brown, textured floor that resembles wood. The background is a clear blue sky with a slight gradient. The sphere has a soft shadow on the floor.

POV-Ray では, 空の名前は以下のものが定義されています. 色々, 試してみましょう.

S_Cloud1

S_Cloud2

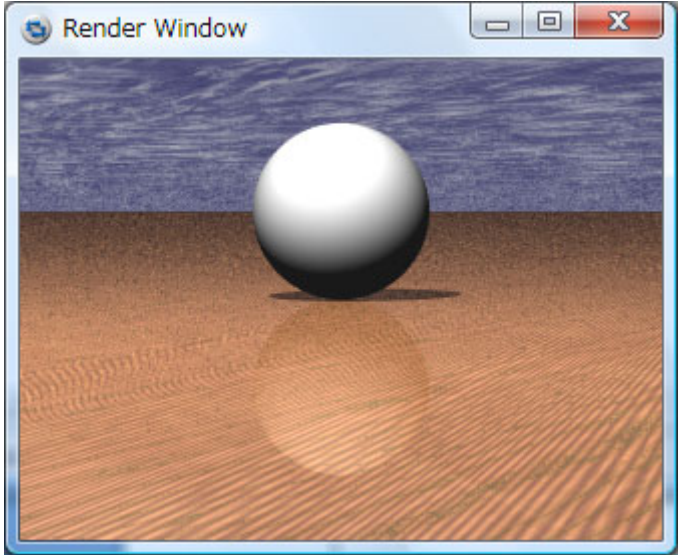
S_Cloud3

S_Cloud4

S_Cloud5

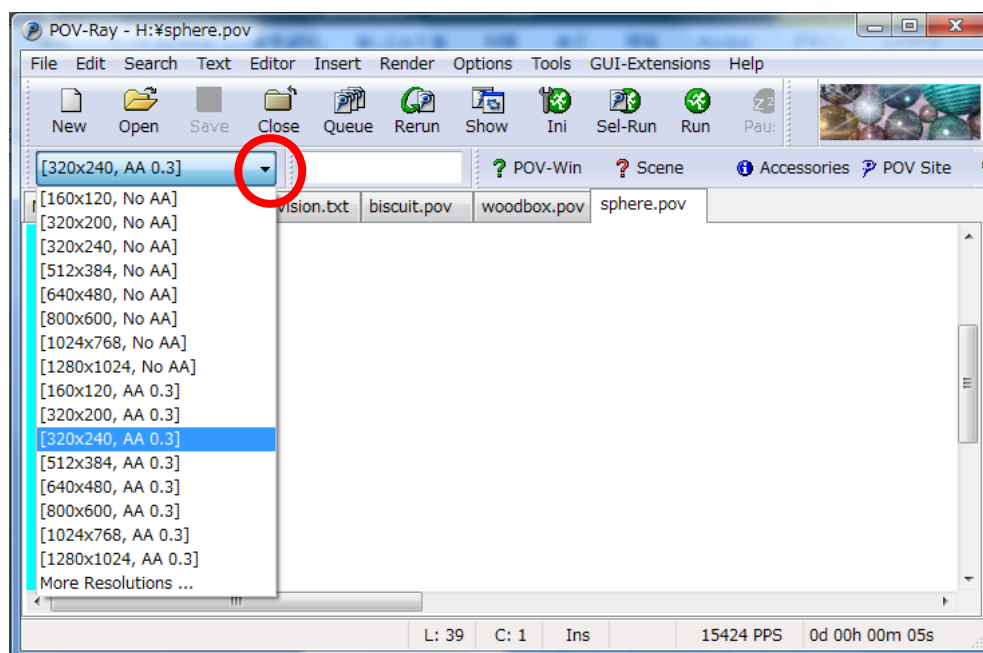
(2) 雲の設定

雲は, object の中に O_Cloud1 または O_Cloud2 を使って作成します. この例では, 空は S_Cloud3 を使いました.

シーンファイル(抜粋)	レンダリング結果
<pre>object{ Plane_XZ texture{ DMFWood6 } } sky_sphere{ S_Cloud3 } object{ O_Cloud2 }</pre>	

8. 画像サイズの変更

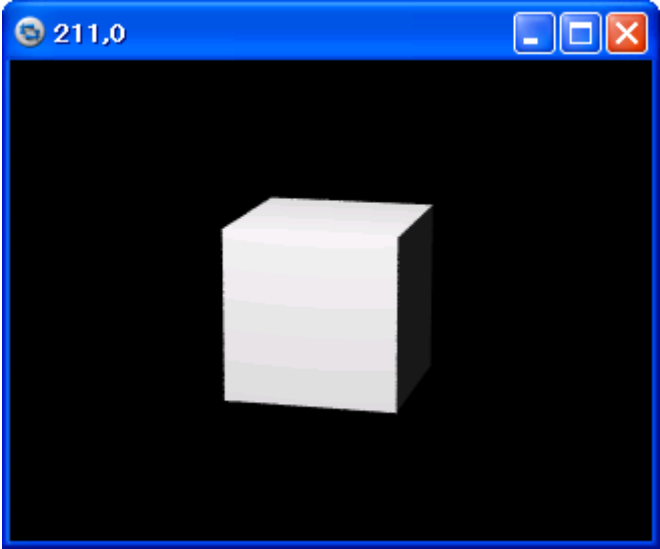
画像サイズの変更は, 下記の [▼] をクリックして行います. No AA の表示は, アンチエイリアシングを行わない設定です. 境界線がギザギザに表示されますが, 表示は早くなります. AA は, アンチエイリアシングを行う設定です. 境界線が滑らかに表示されますが, 表示は遅くなります. 目的に応じて, 選んでください.

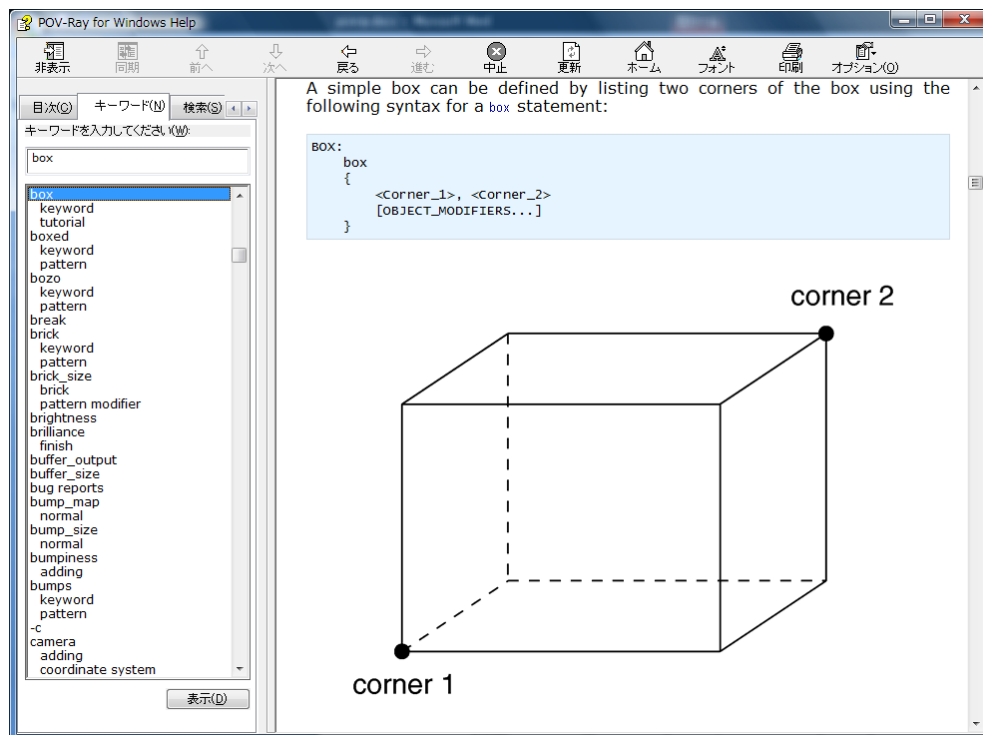


9. その他

(1) box

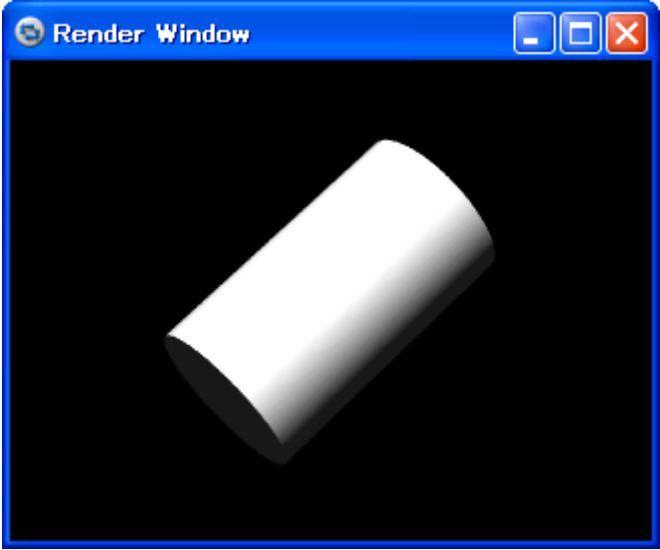
box は、箱のいずれかの頂点と、箱の中心を通過して反対側にある頂点を指定します。

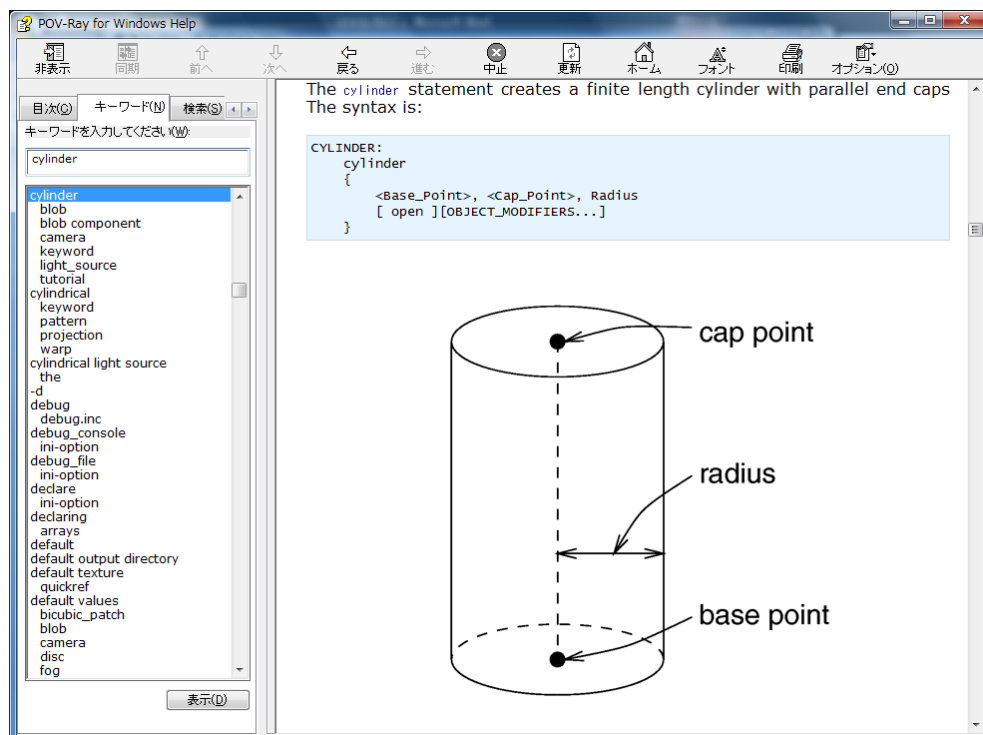
シーンファイル	レンダリング結果
<pre>#include "colors.inc" #include "shapes.inc" #include "textures.inc" camera{ location <5, 5, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } box{ <-1, -1, -1>, <1, 1, 1> pigment{ color White } }</pre>	



(2) cylinder

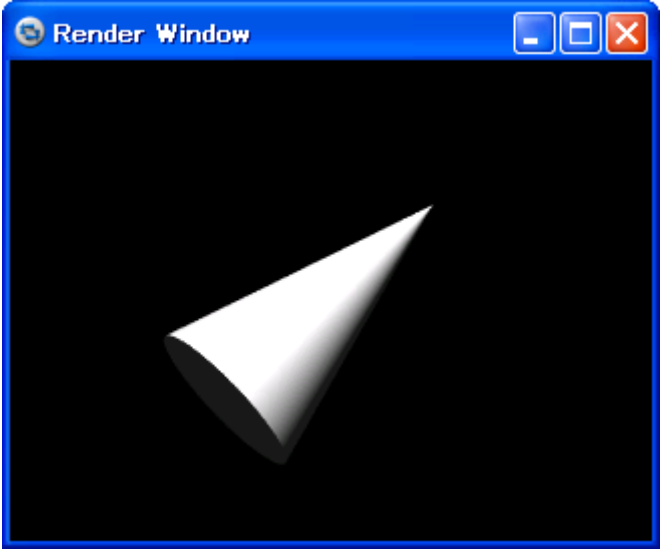
cylinder は、底面の中心座標、上面の中心座標と半径を指定します。

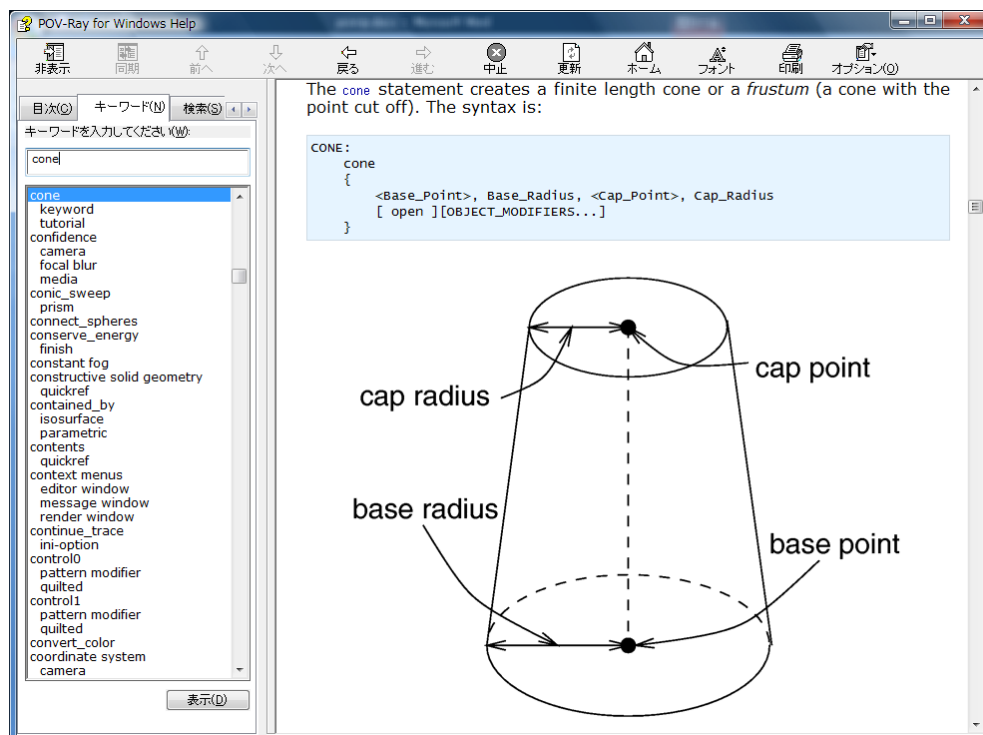
シーンファイル	レンダリング結果
<pre> #include "colors.inc" #include "shapes.inc" #include "textures.inc" camera{ location <5, 5, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } cylinder{ <-1, -1, -1>, <1, 1, 1>, 1 pigment{ color White } } </pre>	



(3) cone

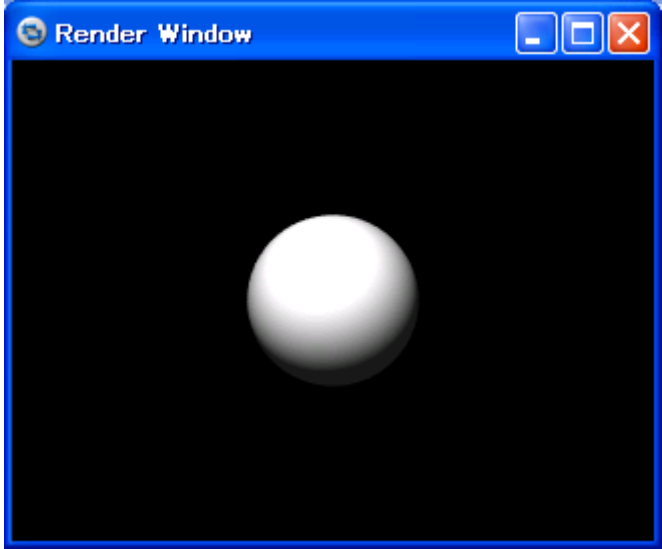
cone は、円錐の底の中心座標と半径、上面の中心座標と半径を指定します。

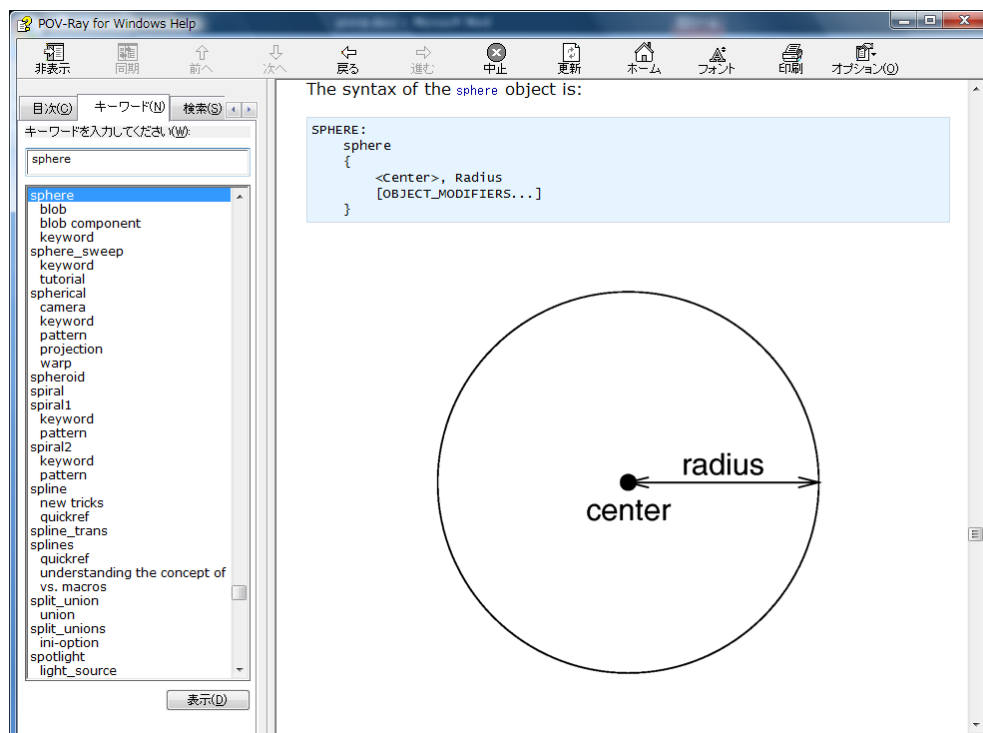
シーンファイル	レンダリング結果
<pre> #include "colors.inc" #include "shapes.inc" #include "textures.inc" camera{ location <5, 5, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } cone{ <-1, -1, -1>, 1, <1, 1, 1>, 0 pigment{ color White } } </pre>	



(4) sphere

sphere は、球の中心座標と半径を指定します。

シーンファイル	レンダリング結果
<pre> #include "colors.inc" #include "shapes.inc" #include "textures.inc" camera{ location <5, 5, -20> look_at <0, 0, 0> angle 20 } light_source{ <0, 10, -10> color White*2 } sphere{ <0, 0, 0>, 1 pigment{ color White } } </pre>	

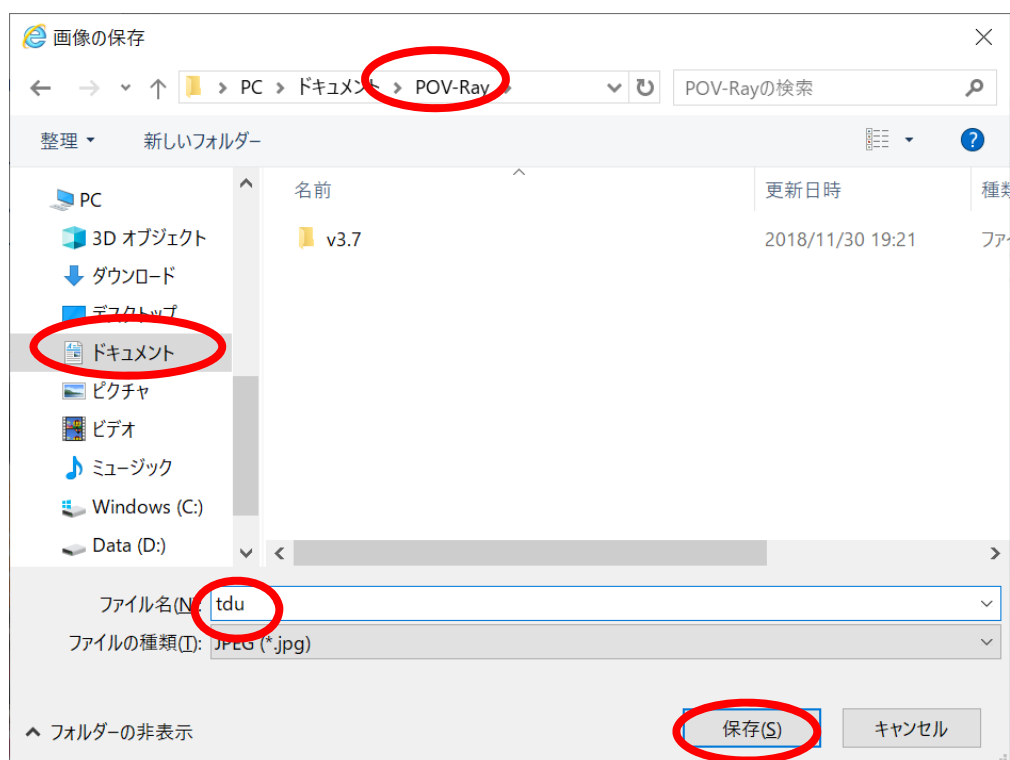


10. 画像の貼り付け

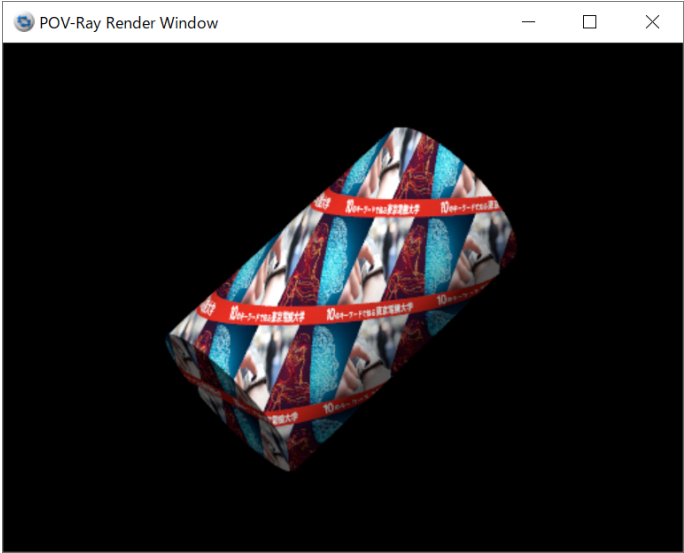
物体の表面に、画像を貼り付けることができます。画像は、本学の web ページから入手することにします。IE を起動し、好みの画像が表示されたら右クリックし、「名前を付けて画像を保存」をクリックします。



先ほど保存したシーンファイルと同じフォルダ、「ドキュメント」「POV-Ray」に保存します。ファイル名は必要に応じて変更してください。ここでは、「tdu」にしました。

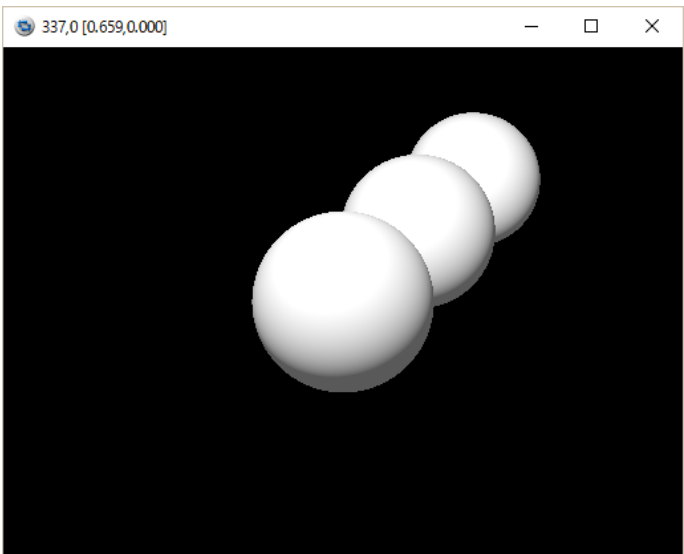


色の代わりに画像のファイルを指定します．なお，PNG 形式の場合は jpeg "tdu.jpg" の部分を png "tdu.png" に，GIF 形式の場合は gif "tdu.gif" に，BMP 形式の場合は bmp "tdu.bmp" にしてください．

シーンファイル(抜粋)	レンダリング結果
<pre> cylinder{ <-1, -1, -1>, <1, 1, 1>, 1 pigment{ image_map { jpeg "tdu.jpg" } } } </pre>	

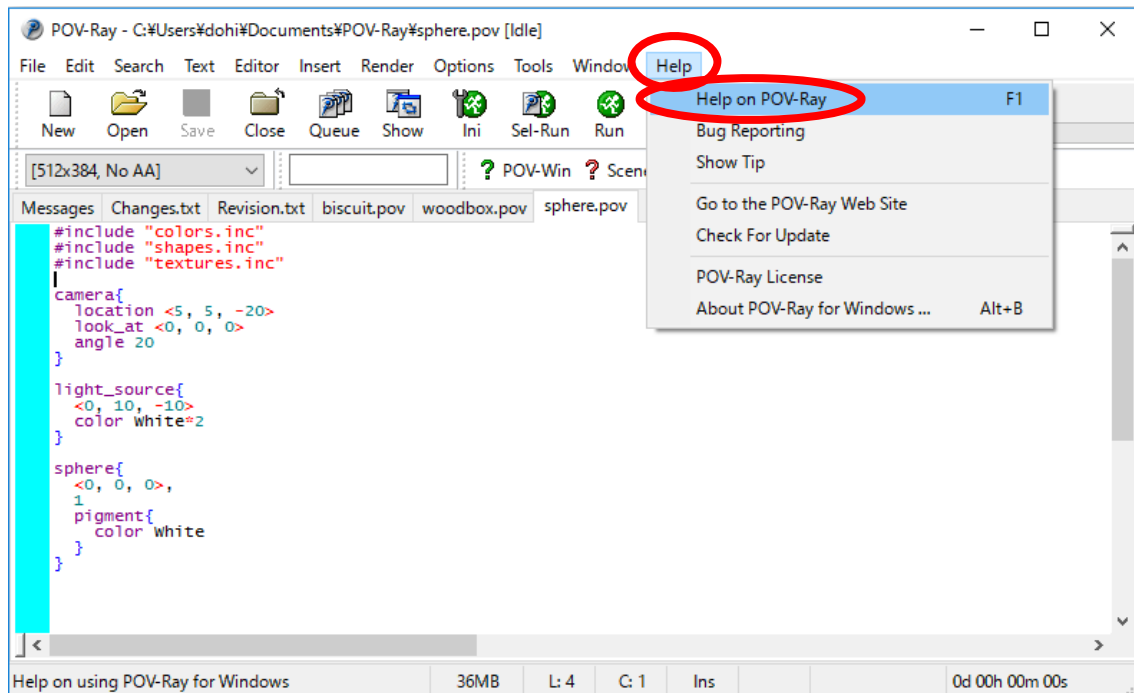
11. 簡単な繰り返し

物体が表示できるようになると，繰り返し(for 文)を使って，複数の物体を表示できるようになります．簡単な繰り返しの例を，以下に示します．for 文の制御変数 i の値が，0 から 2 まで 1 ずつ増加します．#for (i,0,2)と#end で囲まれた部分が繰り返しの対象になります．この結果，Z 軸の正の方向に物体が 3 つ表示されます．なお，プログラミングの基本は，コンピュータプログラミング I の授業で教わります．

シーンファイル(抜粋)	レンダリング結果
<pre> #for (i,0,2) sphere{ <0, 0, 0 + i * 4>, 1 pigment{ color White } } #end </pre>	

12. ヘルプの活用

「Help」「Help on POV-Ray」の順にクリックします。



目次，キーワード，検索の機能を活用しましょう。

