

3D Computer Graphics

NOTE: Copy and paste may not work from your web browser. Please save this file to your computer and open it normally (by double clicking the downloaded file) before continuing with the assignment.

Objectives:

After completion of this assignment, you should be able to:

1. Describe the ever increasing importance of Computer Graphics in our Information Society, and several common applications.
2. Understand the ray tracing model of the 3D rendering process.
3. Create and render simple 3D scenes in Povray using its scene description language.

Introduction:

3D graphics is one of the most useful branches of computer science today. Its applications include:

- Scientific Visualization
- Human Computer Interaction
- Architectural and mechanical design
- Medical imaging
- Simulation
- Geographic Information Systems
- Photo realistic animation for science, commerce and entertainment

We will be learning some of the basic ideas behind 3D Computer Graphics in the process of creating our own simple scenes using the open source ray tracer called POV-RAY. Povray is capable of creating professional quality images. Here is one example:



Bonsais - *Jaime Vives Piqueres*

This is not a photograph of an existing scene. Those are not real Bonsais. The scene is generated entirely in the computer. Here are several other examples (all to be found at the official Povray site: <http://hof.povray.org/>).



[Marbles](#) - [Christoph Hormann](#)



[Car Study](#) - [René Bui](#)



[Patio](#) - [Jaime Vives Piqueres](#)



[Warm Up](#) - by Norbert Kern



[The Eye](#) - [Christoph Hormann](#)

Since we have only a short time with this subject, we will be building much simpler scenes. If you are wondering how it is possible to create the last two realistic landscapes, we cover this and many other interesting ideas in our course CSIS 3400: Computer Graphics Programming.

First -- a very quick overview of the image creation process:

3D scenes are typically modeled as polygonal approximations, or in terms of surface equations, such as the equation of a sphere. There is a viewpoint or camera, a viewing screen or image and the world to be modeled. The task of determining what color to assign each pixel in the image is called "rendering." The method of rendering we cover in this module is called ray tracing. Ray tracing is a computational method for rendering 3D images.

In ray tracing we figure out the color of a pixel by trying to determine the prehistory of the ray that passes through the pixel to the viewpoint. Working backward in time, we first have to find the polygon hit by the ray. What light sources illuminate it? What are its light scattering properties? If it has a specular or refractive component we have to repeat the process as if this intersection point were the viewpoint. Ray tracing is computationally intensive but is capable of producing photorealistic images.

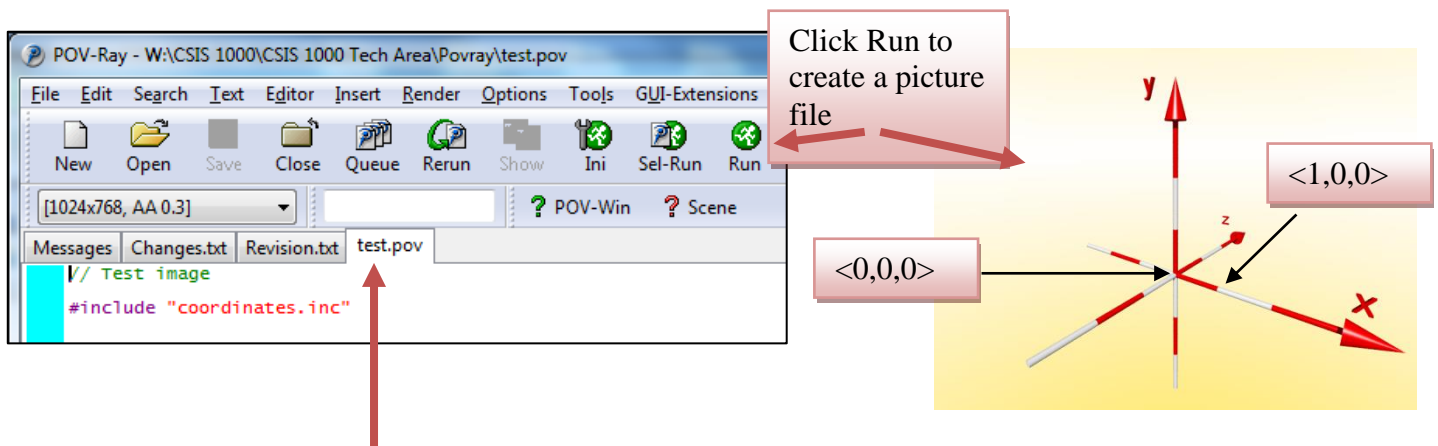
Instructions for Creating Your Own 3D Scenes

The files needed for this assignment were unzipped when you did the first assignment in this class. These files can be found in your CSIS 1000 → Technology → Intro_to_3D_Computer_Graphics folder. If for some reason you do not have this folder, do the following:

- Use a web browser and navigate to this course's main web page.
- Click the calendar link and then select the month when you started this course.
- Click the Intro to CSIS 1000 link on the calendar day.
- Download the CSIS_1000_Files.zip file.
- Right-click the downloaded CSIS_1000_Files.zip file and select Extract All. Follow the instructions in the pop up window to unzip the file. You should now have a folder called CSIS_1000_Files.

Installing Povray and Changing Default Options:

- Povray is installed on campus, so skip this step if you are on a campus computer. Install Povray. We are going to be using the 3.6 version of Povray available at this link: <http://povray.org/download/index-3.6.php> Download the 32 bit version for windows.
- Run Povray. On campus click the SUU Programs folder, then click CSIS → Povray for Windows. If you installed it at home, launch it like you would any other program.



- Open the `test.pov` file in the Intro_to_3D_Computer_Graphics folder and press the **Run** button. Several popup windows may appear. You can safely click the "Do not show this message again" option. One of the windows should display a left-handed coordinate system like the picture above. Notice that there is an x, y, and z axis. Povray refers to the coordinates using the following format $\langle x, y, z \rangle$. For example, a point at the center of the graph, sometimes referred to as the origin, has the coordinates of $\langle 0, 0, 0 \rangle$, and a point 1 unit to the right on the x axis is $\langle 1, 0, 0 \rangle$.

a. Troubleshooting:

- If the `coordinates.inc` line in your Povray window is highlighted in yellow, you do not have the Povray files downloaded and unzipped correctly. Follow steps A through E above.
 - If a window pops up but no image is displayed, you probably have run out of space on your F: drive. Delete some unimportant files off your F: drive and try again. Alternatively, save all the Povray files onto a USB flash drive and try again.
- Change the Povray option located at the top left from [512x384, No AA] to [1024x768, AA 0.3] by clicking the down arrow. This option tells Povray to create an image that is 1024x768 pixels which is anti-aliased (a sharp, clean image).

Creating a Scene in the Povray Scene Description Language:

5. You create a scene in Povray by describing that scene using a Scene Description Language. Open and run the sphere.pov file. Notice the extra text shown in the text box below. This text describes a yellow sphere with a radius of 1 centered at the origin.

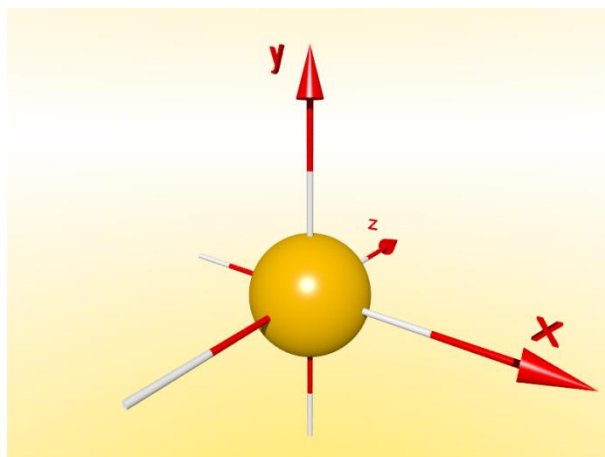
```
sphere{ <0,0,0>, 1.0  
  
    texture { pigment{ color rgb<1.00, 0.55, 0.00>}  
                finish { diffuse 0.9 phong 1.0 reflection 0.00}  
            }  
  
    scale<1,1,1> rotate<0,0,0> translate<0,0,0>  
}
```

The first line uses the keyword sphere, indicating to Povray that a sphere is desired. The <0,0,0> indicates the position of the center of the sphere. The 1.0 is the radius of the sphere. To shrink the object, you could change this number to 0.5.

The next grouping of text indicates the texture that the sphere should have. The important part for this example is the “color rgb<1.00, 0.55, 0.00>” line. RGB stands for Red, Green, and Blue, and indicates the color of the object. A red sphere would be <1.00, 0.00, 0.00> and a blue sphere would be <0.00, 0.00, 1.00>. When you combine red and green light (not paint), a yellow color is produced. Therefore, <1.00, 0.55, 0.00> indicates a yellow sphere.

The last grouping of text contains the scale keyword. This option tell Povray if the object should be larger or smaller in the x, y, or z direction. For example, if you wanted to shrink the object, you could change the 3 components of the scale option to <0.5, 0.5, 0.5>.

The following picture shows the result of running the sphere.pov file with Povray.



6. Do the following experiments:
- Change the radius of the sphere to 2.0 and run the file. The yellow sphere is now twice as large.
 - Change the color of the sphere to green <0.00, 1.00, 0.00> and run the file. The yellow sphere is now green.
 - Scale the sphere in the z direction to 0.2, <1, 1, 0.2>, and run the file. The sphere is now 0.8 units shorter in the z direction and looks like a frisbee.

7. If you would like to see other examples of shapes, open and run the following files. Then change a few numbers like you did in step 2 and run again.
 - cone.pov
 - torus.pov
 - lathe.pov, note that this is an advanced shape
 - sor.pov, note that this is described by a surface of revolution

Inserting Pre-Built Objects into your scene:

8. Click the New button, and then save the Untitled file as Firstname_Lastname_2_spheres.pov in your Intro_to_3D_Computer_Graphics Folder.
9. Insert the Checkered Floor object by selecting Insert → Scene Templates → Checkered Floor. This povray code illustrates many new ideas, such as a camera.
10. Verify that the picture size is [1024x768, AA 0.3], and press the Run button. A picture of a sphere on a checkered floor appears.
11. The camera is the viewpoint of the scene. Change the camera location by changing the associated location text in the angle brackets (< >) and press the Run button again. If you can no longer see the ball and the checkered floor, change the location again until you can.
12. The **plane** object text is used to describe the checkered floor. The wavy appearance towards the top of the floor, sometimes called aliasing, is caused by the repeated checkerboard pattern. Anti-aliasing is the removal of jagged edges by coloring the neighborhood of each pixel. Increase the scale of the **plane** to a number larger than 0.5 and run the file. This will increase the size of the squares on the floor and somewhat reduce the wavy appearance.
13. The **sphere** object text (not **sky_sphere**) is located at the bottom of the text file. The text **0.0, 1** just below the word **sphere** is short hand for **<0, 0, 0>, 1** and indicates the position of the sphere and radius. Change the **0.0, 1** text to **<0, 0, 0>, 1** and run the file to verify that you did not make a mistake in your typing. The same picture should display.
14. Select all the text from the word **sphere** (not **sky_sphere**) upto and including the bottom '}' in the file. Copy and paste the selected area 1 time at the bottom of the file. There are now 2 areas of text labeled **sphere** in your file.
15. Change the position of each sphere from **<0,0,0>** to a different position, and run the file again. If there are less than 2 spheres in the picture, make smaller number adjustments until both spheres are visible when you press the Run button.
16. Paste the following text at the very end of your document

```
text {  
  ttf "crystal.ttf", "Firstname Lastname", 0.5, 0  
  translate<0,0,0>  
  texture { pigment { color rgb <0.8,0.8,0.3> }  
    finish{ diffuse 0.3 ambient 0.0 specular 0.6  
      reflection { 0.8 metallic } conserve_energy  
    }  
  }  
}
```

17. Replace the **Firstname Lastname** text you just pasted into your document with your First and Last name and press the Run button. You should now see your name in the image.
18. Adjust the **translate** numbers so that your entire name appears in the image. **Remember**, the first number is left and right, the second number is up and down, and the third number is in and out. You should have a picture similar to the one below.



Using the skills you have learned, do something else interesting to the image:

19. Using the skills you have learned, do something else interesting to the image. You can change color, insert shapes from some of the other povray files that came with this assignment, change size, etc, just make sure that what you do is obvious so we can see it when we grade your assignment.

A note about image files:

Everytime you press the Run button, the picture that displays is automatically saved in bmp (picture) format with the same name as the .pov file and in the same folder as the .pov file. For example, the picture for Firstname_Lastname_2_spheres.pov has been saved as Firstname_Lastname_2_spheres.bmp in the Povray folder.

20. Submit the following file through Canvas for grading.
 - a. Firstname_Lastname_2_spheres.bmp

Your file will be graded as follows:

21. be 1024x768. (5 points)
22. have 2 spheres placed anywhere in the image (10 points)
23. have your first and last name. (10 points)
24. have something else interesting done to your image. (25 points)

NOTE: Upload the .bmp file and NOT the .pov file.

F: Drive Storage Issues:

The .bmp files generated during this assignment may take up about ¼ of the space on your F: drive. You can safely remove all .bmp files from your F: drive except the 2 you submitted for your assignment. If for some reason you need to recreate a .bmp file, you can do so by using your saved .pov file and PovRay.

Exploring further:

There are many excellent resources for those wishing to learn more on this topic. An excellent place to start is the set of on-line tutorials by Frederick Lohmeuller, from which several of these

examples were adapted. You can follow these links by pressing the Ctrl button and clicking on the blue text below.

[Lohmueller's Povray Tutorials](#), [Official Povray site](#), [Official Povray Tutorial](#), [Povray Reference](#)

Key terms:

Rendering

Ray tracing

Scene description language

Coordinate system

Camera

Light source

RGB Color

Texture

Box, cone, plane, torus

Transformations: rotation, translation

Anti-aliasing