Introduction to 3dsMax

TNM061/TNGD25 Lab 3: Lighting and rendering

Contents

1 Light sources **2**
1.1 Simple light sources 2
1.2 A directional light 4
1.3 Choice of Renderer 4
1.4 Area lights 5
1.5 Secondary light 5
1.6 Render settings 6
1.7 Skylight 8
1.8 Exposure Control 9
1.9 Daylight System 9

2 Reflection and refraction **10**
2.1 Test scene 10
2.2 A wine glass 11
2.3 A glass material 13
2.4 Render settings 14
2.5 Caustics 14

3 Assignments for grading **16**

1 Light sources

Lighting is an important part of computer graphics, both aesthetically and in terms of technology. Good lighting is very important for the overall result, and bad lighting can ruin an otherwise well-made scene. Real-world lighting for photography, film and stage is a difficult art which takes a long time to master, and doing it well requires practice, talent, patience and a critical eye for evaluating the result. The same goes for lighting in computer graphics, and this is by no means an introduction to the aesthetics in lighting design. What is is, though, is am introduction to the lighting tools that are available in typical modern 3-D software. Even if you won't become a master of creating a good looking lighting after only a four hour lab session, you will at least gain some experience in using the common tools of the trade, and hopefully gain some insight into the methods, their possibilities and their limitations.

To start off, we need something to illuminate. Create a simple scene from boxes that looks something like the image below. Create a material with only diffuse reflection (*Specular Level* = 0), and set its diffuse color to light gray. Apply the material to all objects in your scene.

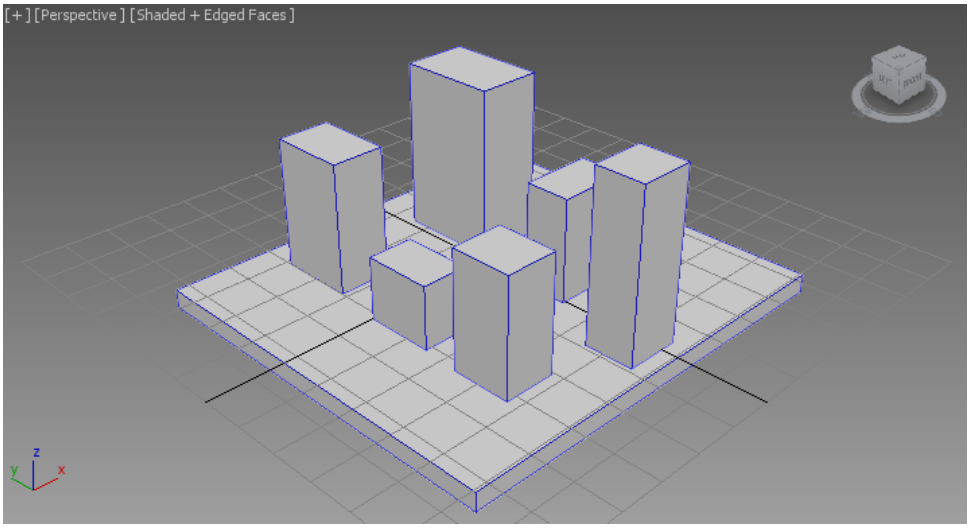

Figure 1: A simple scene for lighting experiments

Render. The result looks flat, doesn't it? If you don't you create a light source in your scene, 3dsMax creates one for you that is very simple. For one thing, it doesn't cast any shadows. Its purpose is simply to illuminate a scene which would otherwise have been pitch black, before you have created any light sources. When you add a light source of your own to the scene, this boring automatic light source is deactivated.

1.1 Simple light sources

Now, create a light source of type *Target Direct*. "Direct" in this case doesn't mean what it usually does. It's a rather inappropriate contraction of the word "Directional", meaning a light source that has parallel rays of the same direction everywhere in the

scene. Light sources that are far away, like the Sun, can be approximated by this kind of light.

Set the light source to illuminate your scene slightly from the side and above, and set the size of the "Light cone", referred to as the *Hot Spot* in the *Modify* tab, so that all objects in your scene are covered by it, except possibly for the floor/ground. To see where the light source is pointing and how large its area of influence is, you can click the title of any viewport, where it says things like "Right", "Front" or "Perspective", and set that viewport to look in the direction of the light source from its current position. The controls at the lower right, those that rotate and zoom the view, will then move and rotate the light source and change its spread. Try it! This is often a very handy way of setting the direction of a light source, even if it may appear a little strange and backwards at first.

When you have placed and adjusted the light source, render again. Note that the new light source doesn't cast any shadows either. You have to activate them for each light source. In the *Modify* tab for the light source, check the box *Shadows - On*, and render again. The shadows might be too crisp and too dark, and the objects might cast shadows on each other in an ugly manner, but at least the shadows are there, which is a significant improvement.

The kind of shadows which are used unless you ask for something else are *Shadow Maps*. It's a slightly old-fashioned but quick way of creating shadows, and it can be good enough in many cases.

A *shadow map* is a depth image, a gray-scale image where each pixel represents not a color, but the *distance* from the camera to the closest object. Shadow maps are rendered in a pre-processing pass from the vantage point of each of the light sources, and during rendering the information can be used to determine whether a point is illuminated by a certain light source. The surface that is illuminated is simply the one closest to the light source, which is the one that is "visible" in the shadow map. Surfaces that are further away are in shadow. (For technical details on shadow mapping, we refer to other sources, like a computer graphics textbook. There is a good article on the subject on Wikipedia as well.) The depth image for the shadow map has a limited resolution, which will become apparent if you zoom in on the scene. To get good results with shadow maps, you need to take care to set the resolution of the shadow maps high enough for the views you want to render.

In this session, we will study light and shadows in detail, so we want to use a more modern way of creating shadows, which is also a better fit with the renderer we will use later on: *Mental ray*. You can do slightly different things with different kinds of shadow algorithms. In the *Modify* panel for the light source, Under *General Parameters* → *Shadows*, change the type to *Ray Traced Shadows* and render again. Do you notice the difference? If not, zoom in on the scene to watch a close-up view of the edge of a shadow.

One advantage with shadow maps is that they make it relatively easy to create blurry shadows. Ray traced shadows become crisp, so crisp that they may be perceived as hard and unrealistic. We won't go into detail on how to blur shadows using a shadow map, but we will describe how extra computations can be used to compute blurry, soft shadows even with ray tracing. Modern computers are fast, and it is not always a wise choice to pick the fastest rendering method. Ray traced shadows are more general than

shadow maps, and can yield a better looking and more realistic result without the 3-D artist having to put in a lot of manual work. However, in cases where you can't use ray traced rendering, like in real time rendering for computer games, shadow maps are still a common and very important tool.

1.2 A directional light

Directional lights simulate light coming from far away, far enough that the rays can be assumed to be parallel. One light source with that property is the Sun, which is distant enough that every point on Earth has more or less the same direction for the sunlight. (Of course, the light has different directions relative to the ground in different places and at different times, but that's because the Earth has a curved surface and rotates relative to the Sun.) Other light sources are often close enough that their direction changes across the scene. A light source simulating that situation is *Target Spot*. Replace your *Target Direct* with a *Target Spot* and study the difference. You don't have to delete the old light source and create a new one - you can just change the type of the existing one in the *Modify* panel. The light rays will now diverge, meaning that they emanate from one point and spread out as they move further away from the light. The shadows will point away from the light source, in slightly different directions depending on where they are. The effect is stronger if you place the light source close to the objects it illuminates.

To mimic reality, the light should also drop in intensity with distance, but there are some problems with getting the intensity right if you do that. Try it by enabling the effect that's called *Decay* for your spotlight. Pick the option that should be correct according to physics, namely *Inverse Square*, making the intensity drop with the square of the distance. You will probably see a scene that looks a lot darker. Increase he *Multiplier* to ramp up the intensity until it looks OK. As you can see, there is a big difference between objects that are close to the light source and those that are more distant, and it looks a little too dramatic. The virtual camera gives you stronger contrast than a real camera or human eyes. Instead, you can choose "Inverse", light that decays proportionally to the distance, un-squared. Strictly speaking, that is just plain silly in terms of physics, but it might look better. Try it! You will probably need to change Multiplier to have just enough light, not too much and not too little.

1.3 Choice of Renderer

For a default installation of 3dsMax, the preset renderer is *Default Scanline Renderer*. It's rather outdated and lacks support for many modern features, but it's also very fast, and its image quality can be sufficient for simple purposes.

The remainder of the exercises for this lab session will use a different renderer: the more modern *Mental ray*. It's considerably slower, but it has much better support for realistic lighting and shadows. Under *Render Setup*, the button with a teapot and a gear to the far right in *Toolbar*, set *Renderer:* to *NVIDIA mental ray*.

If you are using 3dsMax 2018, you won't find Mental ray installed by default, but you can download it for free from `nvidia.com`. The bundled renderer for 3dsMax 2018 is *Arnold*, which is very different from *Mental ray*. Arnold requires you to use its own custom lights and materials, and it is *very* slow. Feel free to try it if you like, but we

still recommend using Mental ray to learn the fundamentals of 3-D graphics rendering. Arnold is somewhat easier to set up, but it creates images with lots of noise. When used correctly, Mental ray can actually create better looking images in less time than Arnold.

1.4 Area lights

Real world lights don't cast perfect, crisp shadows. This happens because they are not a single point, which means that they will be partially hidden from view at the boundary between light and shadow. Most shadows are "soft", not "hard." At the boundary between light and shadow, there is a *penumbra* region (literally, "half-shadow"). Penumbra effects are simulated by using *Area light sources* in the scene.

Create a new light source of type *mr Area Spot* (where "mr" is short for "mental ray") and remove or turn off the light source you already have in your scene. In the rollout *Area Light Parameters*, set the shape and size of the area which the light source is supposed to cover so that it becomes reasonably large compared to the objects in your scene. Set *Height* and *Width* to about the same values as one of your boxes. Also, don't place the light too far away. You will see a representation of the area light source in the viewport. Render. Watch how the shadows are crisp near the objects that cast them, but more blurred further away.

The rendering of a penumbra is done by looking not at a single point on the light source, but at several points across its surface, acquiring several *samples*, and counting the number of points that are visible from the object. The number of samples can be controlled by the setting *Samples* in the rollout *Area Light Parameters*. In general, more sample points are needed for large lights and lights which are close to objects – something which is referred to as (*soft light*).

1.5 Secondary light

Perhaps you noticed that even the shadowed parts of your scene are slightly illuminated. Where does that light come from? The answer is that this is what is called *secondary light* or *indirect light*, light which bounces off surrounding objects and then ends up on surfaces that are not illuminated directly. The methods behind this are complicated and require lots of computations, but computers are now fast enough to handle it even for routine rendering. This is a fairly recent development. Methods that are used routinely today were impossibly complicated for production work only a decade ago. Secondary light in Mental ray is computed by two methods: *Final Gather* and *Photon Mapping*, methods that are actually not all that old.

Final Gather, simply put, is a pre-rendering pass where you first compute an image with only orect illumination, oftran at a lower resolution, and then use that information in a second rendering pass to take also indirect light into account. The process is an approximation, but it's reasonably accurate and creates visually pleasing results.

Photon mapping, again simply put, involves first shooting "photons", light rays, from all light sources in the scene, keeping track of where the rays end up. In the subsequent rendering, you look at direct illumination and indirect light from photons that either passed through transparent objects or reflected off shiny surfaces.

Slightly more detailed explanations of *Final Gather* and *Photon Mapping* is available on Wikipedia, or in the corresponding Help section in 3dsMax.

1.6 Render settings

Now, let's find out what happens if you change the quality settings for the rendering. Keep the rendering window open and look at the controls beneath the image, according to Figure 2. All settings have one thing in common: The image gets better but renders more slowly when you drag the sliders to the right, and it gets worse when but renders more quickly when you drag them to the left. Nothing comes free in 3-D rendering – it is always a compromise between time and quality.

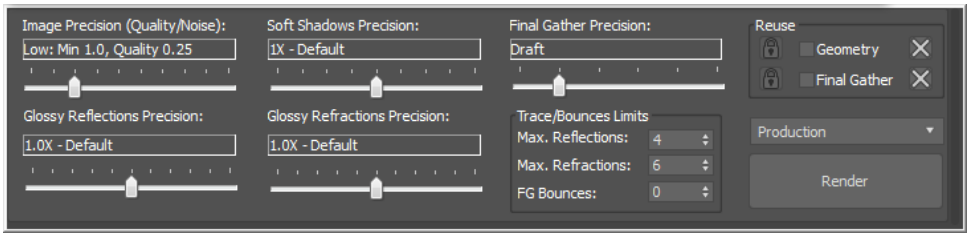

Figure 2: Settings in the Render window

First, experiment with the slider *Image precision*. At the left endpoint of the scale, the image has low resolution and may become noisy. In the middle, some edges may be a little jaggy and the image might contain visible noise. To the far right, the images become very nice, but they also take a very long time to render. Our scene is very simple, but the rendering times can still run out of hand if you push all sliders to maximum quality. The method used here to render pixels is called *adaptive stochastic multisampling*. "Adaptive" means that the renderer has some freedom to save in on samples where not much is happening in the image, e.g. at smooth, evenly lit surfaces, and use more samples in areas where the image is extra busy, e.g along edges. Jittered means that the sample point is not always at the exact center of the pixel, but shifts somewhat between adjacent pixels to hide some sampling artefacts that would otherwise be very visible.

"Stochastic" means that each pixel is sampled multiple times, at slightly different locations, and the final pixel value is the average of the pixel evaluations. For good looking results, very small features (less than one pixel wide) may be broken up and replaced with noise, thereby avoiding crisp defects, which is something our eyes are very sensitive to. The rather unspecific slider "Quality" controls the number of samples that are taken for each pixel. If this is less than 1, the renderer will skip some pixels, assuming (correctly, in most cases) that the image is going to be OK anyway.

If the smallest number of samples is less than 1, the image will be interpolated across some pixels, and the effort will be spent on areas with lots of detail. If "Quality" is "high", each pixel will have at least one sample. What constitutes "good enough" for this slider depends on what the scene is, what it is gping to be used for. You need to experiment a bit to find a good value for your particular task at hand.

The slider *Soft shadows Precision* controls how many shadow rays should be sent to determine how much of a certain light source that is obscured. To the far left, area light

sources are disabled but still strong, casting crisp shadodws. To the far right, lots of photons are sent to probe the scene to "stick" to the surface of objects in the scene, in a pre-processing step before the rendering proper. The setting is a relative scale, with the center value being "1X - Default". Move the slider and study the result. At a lower quality settings, you will notice noise and defects in your scene. Soft shadows are generally difficult to render, but it's hard to manage without them if you want to create good looking images of indoor scenes.

The quality for soft shadows can also be set separately for each light source, by changing the number of *Samples* for the light. The slider in the rendering window sets the number of samples in relation to what was specified centrally. "1X" means to use as many samples as the sample light source setting says, but the factor can be adjusted both up and down within wide limits.

The two sliders labeled *Glossy Reflections Precision* and *Glossy Refractions Precision* control the quality of semi-transparent and semi-reflective surfaces. We are not going to make that kind of material in this lab session, but please remember that these setting are available, in case you need them.

The setting *Final gather precision* is related to secondary light. If you pull this slider all the way to the left, Final Gather is disabled, and shadows will be pitch black. This would almost never be the case in a real world situation, so beware.

Sometimes you can get away without using it, but it's often very good to use at least some *Final Gather*. It doesn't cost a lot in terms of rendering power, and it often improves the image a lot. If you instead pull the slider further to the right, you see more subtle changes in the rendered image. Among other things, there is a more distinct lighter region where two objects meet in an inner corner. (If you don't see it in your scene, make the material lighter.) It looks kind of strange when you look at the sample sphere without any other objects in the scene, but this is how real light behaves. Light is reflected off a surface from other bright surfaces nearby. More light is reflected off a bright surface when it's within reach of another bright surfae that reflects light onto it.

If you want to see the effect of *Final Gather* really clearly, put a saturated, bright color to one of your boxes, render and watch what happens to neutral surfaces around it. You should see at least a faint tint from the colored surface on other objects around it. This is called *color bleeding*. You might need to experiment with the direction of the light to see the effect clearly. Despite being a rather subtle effect, color bleeding can be surprisingly important to the overall realism of a scene.

The spinner at the bottom named *FG Bounces* deals with *Final Gather* (FG) as well. In very diffuse and very bright scenes, you sometimes need to increase the number of bounces for diffuse interreflections. The default setting of 0 does not strictly mean zero bounces, but *one* bounce, i.e. light emanating from a light source bouncing from one diffuse surface and ending up on another diffuse surface. Incrementing this value means that one extra bounce between diffuse surfaces is considered.

The settings *Max Reflections* and *Max Refractions* will be revisited later, when we get to the section on reflection and refraction.

Settings in the box "Reuse" are not covered in this course, but we want to mention why it's there. Sometimes, in a rendered movie sequence, you can save a considerable amount of work bu re-using some information from the previous frame. One of the things you can often save between frames is the information on how indirect light is spreading.

In some cases, it also helps not to recompute all the lighting for a scene for each frame. Sometimes you can speed things up by storing the indirect light (photons) once and then reusing that information for each subsequent frame. You can also reduce the visual noise in video sequences. Have a look in the Help section of 3dsMax if you want to know more.

1.7 Skylight

Not all scenes have a strong, directional light source as its dominant illumination. There is often a strong influence of light coming from many directions. Indoors, ceiling and walls are often a bright color and contribute strongly to the indirect light in the scene. It may be better to simulate that with a diffuse light source rather than rely on *Final Gather* or some other algorithm to handle it. Outdoors, in addition to direct sunlight, there is considerable illumination coming from the sky. On an overcast day, indirect illumination from the sky is the only light source, and the light illuminating a point comes more or less uniformly from all directions where the sky is visible from that point. Because these cases are difficult to handle using ordinary light sources, they have their own type of light source: *Skylight*. Deactivate or remove your previous light sources from the scene and create a *Skylight*. The position of this particular light source doesn't matter – just place it where you think it's convenient.

Render your scene with a Skylight as the only light source and watch how soft the light becomes. Unfortunately, the image also becomes very noisy. (We'll fix that shortly.) THe noise is because Skylight is effectively a very large area light, and it is designed for use as a secondary light, the light from the sky filling in a diffuse hemisphere above while the Sun is doing all the heavy lifting. The visual noise would then only occur in the shadows, where it might not matter all that much. Here, the Skylight is deliberately the only light source, and we want any such defects to stand out. To create a good image with less noise, you need to spend more effort on rendering the Skylight. This can be done in two ways: increase the quality setting in Render Setup window, or change to a different algorithm for the Skylight computations.

To change the settings, press the button *Render Setup* and find the tab *Global Illumination*, see Figure 3, and find the tab *Skylights and Environment Lighting (IBL)*. Increase the *Quality* to 1.0 or even some more, render and watch the difference. Unfortunately, getting rid of the noise altogether requires time. However, there is another option: click the option *Skylight Illumination from Indirect (GI or FG)*. With that setting, simplified and less accurate models are used to compute the skylight, but the result is a lot less noisy. Using this option, the quality of the shadow is controlled by the slider *Final Gather Precision* in the render window.

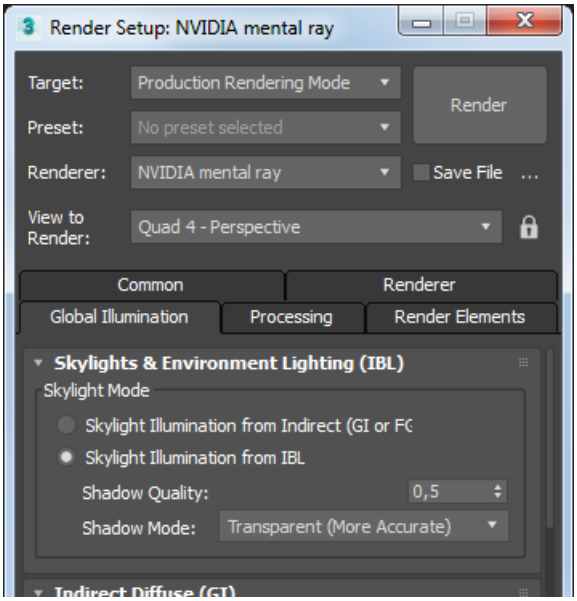

Figure 3: Render settings for Skylight

1.8 Exposure Control

In case the scene is rendered too dark (possibly completely black), or too bright (possibly all washed out and white) during your experiments, you want to edit the exposure control setting for the renderer. In the menu *Rendering* → *Exposure Control...*, you can decide how the renderer should interpret illuminaton values that may vary not only within the range from 0.0 to 1.0, but significantly beyond that and translate them to pixel values for the output image. The settings *mr Photographic Exposure Control* or the similar *Exposure Control* are both reasonable choices.

1.9 Daylight System

Outdoors, you have a combination of light from the Sun and light from the sky. In order to make it easier to create realistic lighting outdoors, there is a ready-made *Daylight System* in 3dsMax. You find it in the menu under *Create* → *Systems* → *Daylight System*, or through the*System* tab in the *Create* panel.

A Daylight System can be a suitable helper tool if you are creating an outdoors scene, because you set concrete parameters like location, date and time instead of explicitly setting the angle, intensity and color for the Sun. Above all, a Daylight System is useful to visualize how light falls (or doesn't fall) in a planned real world environment that hasn't been built yet, or a historic place that no longer exists. Such visualizations often want to show what the building looks like in different seasons and at different times of day. We won't go into any detail on how *Daylight System* works, but it's basically a combination of a *Skylight* and a *Directional light*, where you specify the location and the time of day instead of explicitly setting the color, intensity and angle for the light source. Details on this are in the Help system.

2 Reflection and refraction

The renderer *mental ray* is a so-called *ray tracer*. Ray tracing is a very suitable rendering method for shiny and transparent objects. Because that kind of objects can be important and central to many virtual scenes, we will study reflections and refractions in some detail for the remainder of this session.

2.1 Test scene

Perform a *Reset* in 3dsMax to remove any lingering settings from the previous assignment. Open *Render Setup* and set the rendering to use *NVIDIA mental ray*.

To show reflections and refractions, we need a test scene that can be reflected and refracted in the objects we create. Instead of making a complex scene with lots of objects, we cheat and make a checkered floor. This is a classic background to show reflections and refractions clearly, even though it's not very realistic.

Create a *Plane* of size 100 by 100 units, and reasonably large amount of segments along both axes so we can bend it nicely. Add a *Bend* modifier in the X direction with *Angle* set to -90, and *Limit Effect* activated with *Upper Limit* 60. Then, add another *Bend* in the Y direction with *Angle* -75. Put a material on the object, and set the diffuse color to a map of type *Checker*. Set *Tile* to 12 along both the U and V directions. The result should be something close to figure 4.

For a more clear visual result in the following assignments, change the color of the checkered pattern from the default white and black to something with a little less extreme contrast, e.g. light gray and dark gray, or two reasonably light and not over-saturated colors. On a completely black surface, a shadow won't show, and on a completely white surface it might be difficult to see "caustics" unless you set the rendering to yield pictures that are too dark.

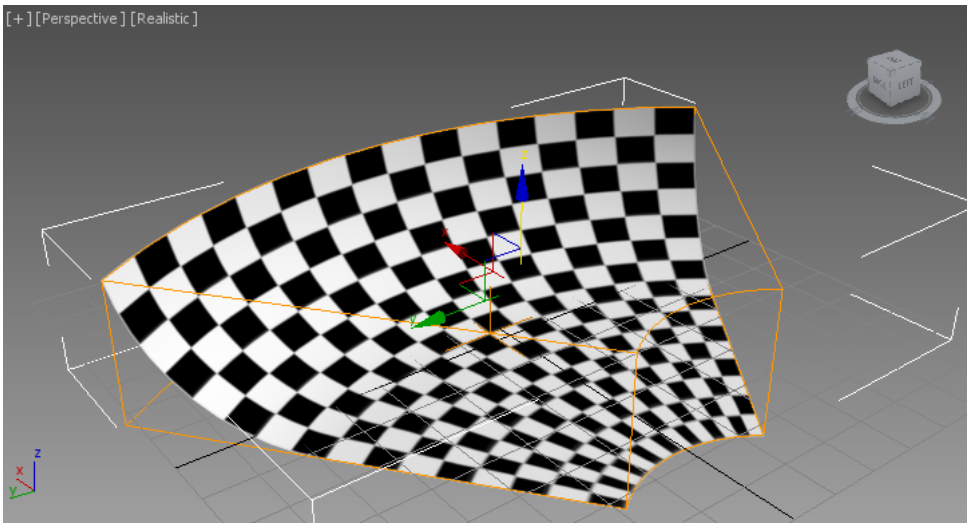

Figure 4: Background for ray tracing experiments

2.2 A wine glass

Let's make an object that has both reflection and transparency. We could choose a simple sphere and design a glass or metal material for it. Many classic articles about ray tracing have been illustrated more or less exclusively with images of spheres on a checkered surfaces. A better option for this particular demo would be a thin-walled *Tube* standing upright on the checkered surface. However, to make things a little more interesting, we will create a wine glass by using the modifier *Lathe*. The modifier can create rotationally symmetric objects.

First, draw a *Line* and make it look at least approximately like the *Line* in Figure5. Draw in the Top view, working in the (x,y) plane. Start by placing all the corner points. Right-click when you set the last point.

Then, line up the two endpoints to have the same x position. There are several wqays to do this, but first of all you need to be able to move individual vertex points in the *Line* In the *Modify* panel for your *Line*, in the rollout *Selection*, pick *Vertex* (the icon with the small dots) to manipulate individual vertices along the line. Select one of the endpoints, do *Align* (alt-A) and click the line itself as the *Target* object. Align along X against *Maximum* (assuming you have drawn the line with the same coordinate directions as the example in Figure 5). Repeat for the other endpoint. Now both vertices are at the same X coordinate, which is iportant for the next step. Unfortunately, the *Align* tool is a little awkward to use for individual points in this manner. If you like, you can instead click the points you want to move and type in exact coordinates in the text boxes at the bottom of the main window. The important thing is that the two endpoints of your *Line* have the same X coordinate before you continue.

To make the line softly curved, right click on each of the points to get a pop-up menu. In that menu, change their type to *Béziér*. For the point where the stem of the glass meets the foot, pick *Béziér Corner* instead, to create a sharp corner. Fiddle with the control points for a while until you are happy with the result. It pays to be patient here, but you can of course go back and edit the line later if you are not satisfied. Keep in mind that a wine glass is very thin, just a millimeter or so, so don't make the wall too thick. That will make your glass look more like a heavy crystal vase than a delicate wine glass.

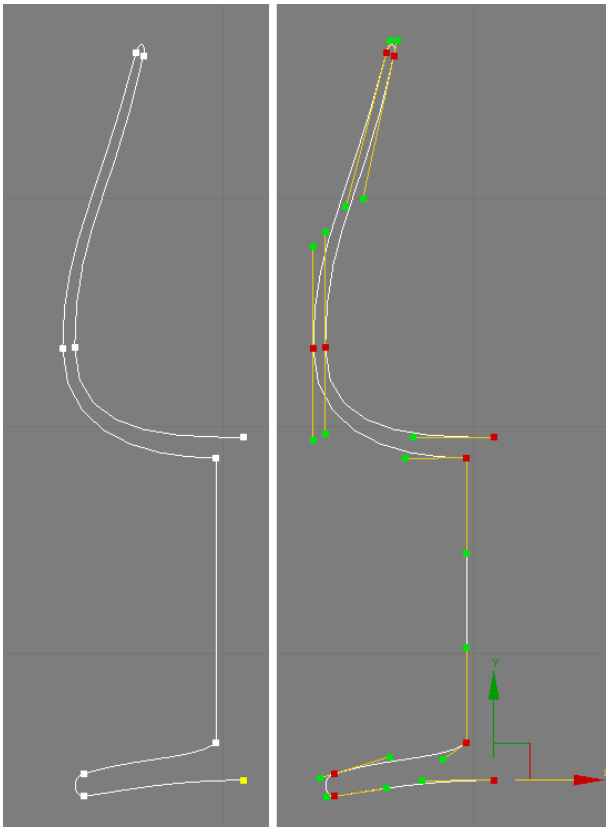

Figure 5: The profile for a wine glass

Now, put the modifier *Lathe* on your *Line*. For the *Direction*, pick *Y*, and for *Align*, pick *Max* Select *Mesh* for *Output* to create an ordinary polygon model. The result should be something along the lines of the left half of Figure 6. If it looks ugly, feel free to try fixing the proportions,, but the aesthetics is not the main point here. If it looks like the right half of Figure 6, you accidentally got the normals pointing in the wrong direction. This depends on which direction you traced your original Line, and it's not obvious to know which direction is "correct". If the problem arises, check the box *Flip Normals* in *Lathe*. Normals pointing inwards instead of outwards is a common problem while modelling, in particular when you create various kinds of swept surfaces like the *Lathe* you are doing now. Also, check the box *Weld Core*, to make the glass better-looking right at the center of rotation.

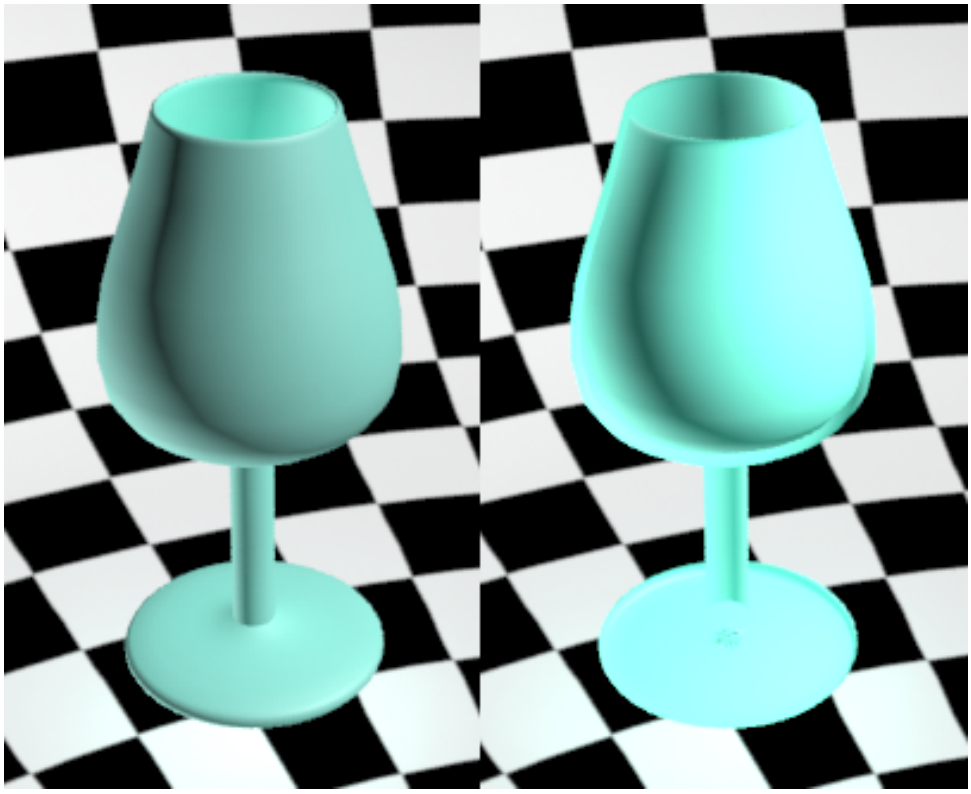

Figure 6: The finished wine glass, with correct normals (left) and inside-out normals (right)

2.3 A glass material

Now, let's make a glass material for the wine glass. First, we are going to do it with a *Standard* material, to better understand what is required of a glass material. You can find ready-made glass materials in the material editor, but you won't learn much from that, so let's wait a while before going there.

Apply a material to your wine glass. Set its type to *Standard*. Set the *Diffuse* color to be completely black. Shiny, clean glass has virtually no diffuse reflection at all. However, set the *Specular Color* to white. For some reason, that color affects reflections made with a *Reflection map*, which is what we are going to do first.

Create a *mr Area Spot* to illuminate your scene, and set up a nice view where you see the glass against the checkered background. Make sure the spotlight throws a shadow, and that you see the shadow in your view.

Glass is transparent, but it has a slightly reflective surface as well. Let's start with the reflection, not because it's the most important property, but because it's the easiest one to make and the one that is simpler to understand. In the *Maps* section of the material, pick *Reflection* and set its type to *Raytrace*. Render. Your glass looks like it was made from polished metal, but at least the reflection is correct.

Glass is transparent as well as reflective. You can simulate transparency in a crude manner by lowering the *Opacity* of the material. Try that. It won't look good. A prominent property of transparent solid materials is that they *refract* light that passes through them, and *Opacity* doesn't address that. Reset *Opacity* to 100 and instead, in the

Maps section, pick a *Refraction* map of the type *Raytrace*. Also, turn off the *Reflection* map by deactivating its check box. We will soon get to why. Render. It should now take noticeably longer to compute the image for the pixels where the glass is, but on a modern computer you should still see reasonably short rendering times, provided you picked reasonable quality settings for the rendering.

Looking closely at your rendered glass, you will notice that there is both a slight reflection in the surface *and* a refraction through it. The reason for this is in the settings in the rollout for your *Raytrace* map. At the bottom, under the heading *Refractive Material Extensions* is a check box *Treat Refractions as Glass (Fresnel Effect)*. Uncheck that and render again. Now the rendering is quicker, but the result doesn't look quite like glass any longer. That's because the reflections are completely gone. You can get them back by reactivating your Reflection map, but there's a problem with that: glass reflects different amounts of light in different angles, and how much light is refracted depends on how much light that is reflected. Reflection and transmission in a material in the real world must sum up to less than 100 percent, or else energy would be created from nothing. It's a little tricky to get two separate *Reflection* and *Refraction* maps to vary their intensities in the manner we want. It's not impossible, but it's a bit of a hassle. That's why this phandy check box is there. It makes it a lot easier to create transparent, glass-like materials. (This includes liquids, like water.)

2.4 Render settings

In the render settings beneath the render window, there is a group of settings called *Trace/Bounces limits*. Reduce *Max Reflections* and *Max Refractions*. If you set *Max Reflections* to 0, you will remove the reflection in the glass surface altogether. If you set *Max Refractions* to 0, the glass will become black, because no refracted ray is sent through the surface to find out what's behind it.

What is the smallest value for *Max Refractions* that gives a reasonably correct rendering of your wine glass? Why this particular value?

2.5 Caustics

There is still one flaw with our image: The glass throws a compact shadow. You can set the shadow to be less dense, but that's not how glass behaves. Light is refracted through the glass and may land partly in the region which would have been in shadow if the glass had not been transparent, but redistributed because of the refraction. In a simlar manner, a curved reflective surface may throw strongly deformed reflexes creating highlights in its vicinity. Such effects are referred to as *caustics*, and unfortunately they are quite troublesome to render even with ray tracing. This is because they are due to light emanating from the light source, hitting a shiny or transparent object and then land on a diffuse surface. Ray tracing in its basic form (the algorithm which is called *inverse ray tracing* or sometimes *Whitted ray tracing* after its inventor Turner Whitted) only handles the opposite case: light lands on a diffuse surface, and the diffuse surface is then viewed through one or more reflections and refractions.

To render caustics in *mental ray*, we need to perform a separate pass of computations before the image is rendered, using a method called *photon mapping*. Unfortunately, this

is an advanced feature which is not supported by the *Standard* material, so to fix this final flaw, we switch to a predefined glass material. This is what most people do, and of course it's the preferred method when using the software for production purposes.

Change the material for your wine glass to *Autodesk Solid Glass*. The default settings for that material are OK for a first test. Render and look at the result. The shadow is no longer black, and it might work at a distance and for a brief glance, but it still looks strange. It's still done by a cheap trick with a lower opacity during the shadow computation. Now we will enable caustics to get a more realistic image.

Right click the wine glass and pick *Object Properties*, or select the wine glass and choose *Edit* → *Object Properties* from the menu bar. Find the tab *mental ray*, and under *Caustics and Global Illumination (GI)*, check the box *Generate Caustics* (see Figure 7, left). DO the same thing also for your light source. Now the light and the glass are involved in the simulation of caustics.

In addition to this, you must also activate the simulation of caustics in the renderer. Because it's an extra rendering pass that takes considerable time to render, it's not enabled by default. Choose *Render Setup* (F10), and in the tab *Global Illumination*, scroll down to the rollout *Caustics and Photon Mapping (GI)*. Under *Caustics*, check the box *Enable* (Figure 7, right).

Render. Now your shadow should look considerably better, but it can still be improved. Scroll further down in *Render Setup* until you find *Light Properties*, and change *Decay* to 1.5 instead of 2.0. Render again. Now, secondary effects like caustics won't decay as rapidly with distance, and the effect is more prominent. It's not necessarily more realistic, but it's more noticeable. Caustics in *mental ray* are simulated in a manner that isn't really physically correct, but it's good enough for most purposes.

To get a more crisp and detailed caustics effect, you also need to increase the number of photons that are sent from the light source. You will find that setting as well under *Light Properties*, and it's labeled *Average Caustic Photons per Light*. BY default, that text field contains the number 20000, which is fairly small for this kind of computation. Increase it at least tenfold and watch what happens with the rendered image, and with the rendering time. This particular scene is rather strange and difficult to render, and you need to spend more effort on caustics than usual to get a good result. Because the caustic effect is so important to the final result, it's not unreasonable to shoot millions of photons and let the pre-computation of the caustics take longer than the actual rendering. Experiment with the number of photons and watch its influence on the rendered image!

(Translation not done for the remainder of the document. Work in progress.)

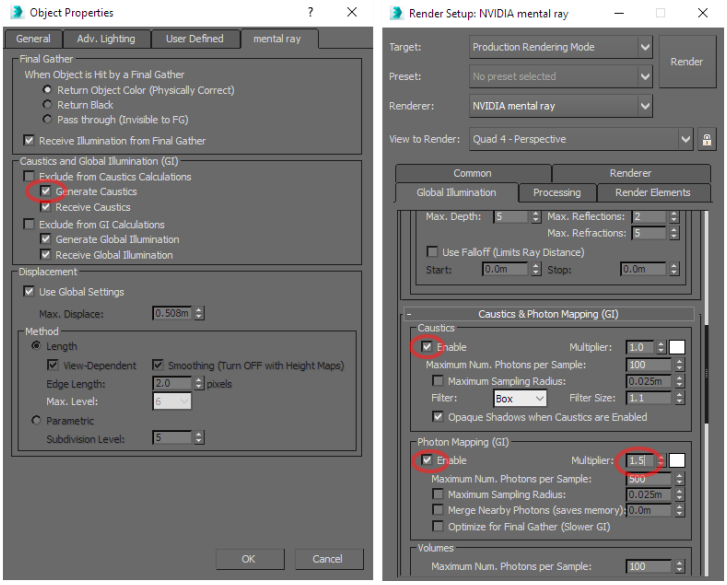

Figure 7: Settings to enable rendering of caustics

3 Assignments for grading

- Study the section on light and shadows, and render an image of your test scene of diffuse boxes to demonstrate soft shadows from an area light source. Render another image with a *Skylight*, and take care not to make the image too noisy. Demonstrate simulated daylight from a *Daylight System* by rendering a few images to show the effect of the light changing for different times of day.

- Render nice images to show the glass you created, one using your own *Standard* material and one using the predefined glass material and caustics. The image should be reasonably realistic both in terms of reflection and refraction, but it shouldn't need to take an hour to render. Use a single light of type *mr Area Spot*. Remember to check the box *Generate Caustics* in *Object Properties* for the light source and the glass alike: for the light source to create any Photons at all, and for the glass to make the photons travel further into the scene.

Show your images to the lab assistant during the scheduled lab session. Good luck, and please ask if anything is unclear or if you want to know more. However, don't forget that you can also find lots of more information online and in the extensive Help section of 3dsMax. And, last but not least: don't be afraid to try things out and experiment on your own!