

Lab 4

IAT 343

K.J. Lee (kla8@sfu.ca)

Due Dates (WebCT > Assignments)

(2/05-midnight)

- Individual: Lab Exercises week 1 – 3 & individual website
- No need to change the color of the objects in Week 1 exercise

(2/12-midnight)

- Team: Storyboard with pitch, voice over describing about your project

(2/05)

- Quiz 1 (mixture of multiple choice and short answer quizzes from the textbook ONLY, not including materials from lecture or lab slides.
 - 20 minutes starting at 6:35
 - Textbook copies available at reserved area in the library

(2/19)

- Individual: Port I. Modeling/Texturing/Photorealistic Rendering

Post your individual website address on the WebCT!
Post your files on your site, DO NOT submit throguh WebCT.

(WebCT > Course Tools > Roster > Students)

<http://www.sfu.ca/~ysh3/iat343/exercise.html>

The screenshot shows a web browser window titled "Blackboard Learning System - Mozilla Firefox". The address bar displays the URL: <http://webct.sfu.ca/webct/cobaltMainFrame.dowebrtc?appforward=/webct/viewMyWebCT.dowebrtc>. The browser's tab bar shows several open tabs: "Google", "SFU Connect: Drafts (366)", "Blackboard Learning S...", "SFU DreamMakers", "Asian Studies: Events", and "ubc korean".

The main content area of the browser displays the Blackboard Learning System interface for Simon Fraser University (SFU). The header includes the SFU logo and the text "SIMON FRASER UNIVERSITY THINKING OF THE WORLD". The navigation bar shows "Build", "Teach", and "Student View" tabs, with "Student View" being the active tab. The course title "IAT343 - E100 Summer11 Animation" is displayed in the top right corner.

The left sidebar contains two main sections: "Course Tools" and "My Tools". The "Course Tools" section includes links for "Course Content", "Assignments", "Discussions", "Mail", "Syllabus", and "Roster". The "My Tools" section includes links for "My Grades" and "My Files".

The main content area displays the "Roster" page. At the top, it says "Your location: Roster". Below this, there is a "Print Page" button. The "View Profiles for:" section has three tabs: "Instructors", "Students", and "Groups", with "Students" being the active tab. Below the tabs, there is a "Page:" dropdown menu set to "All (90)".

The roster displays student profiles. The first profile shown is for a student with the following information:

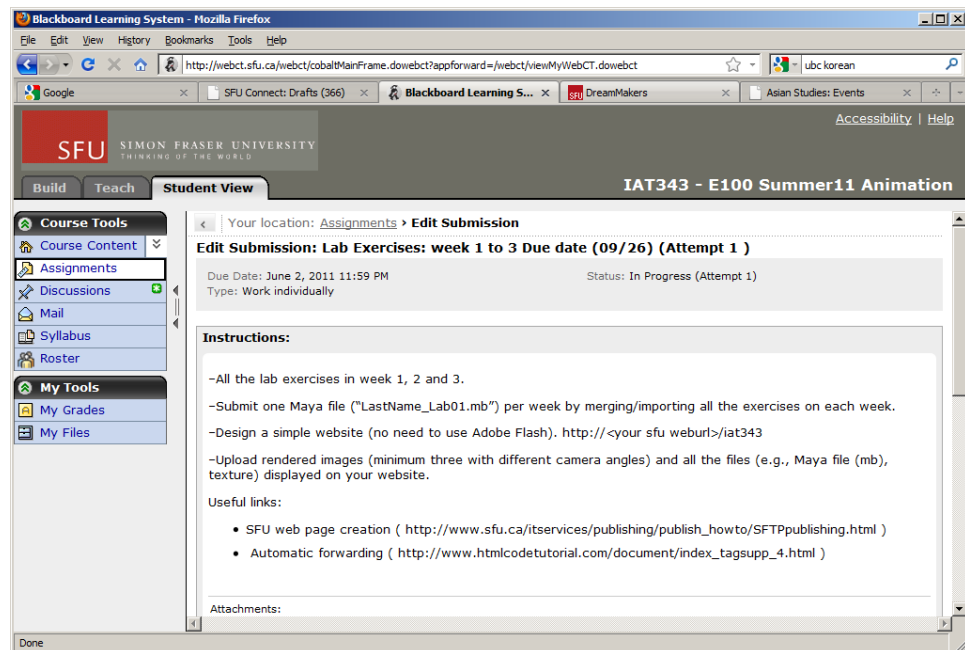
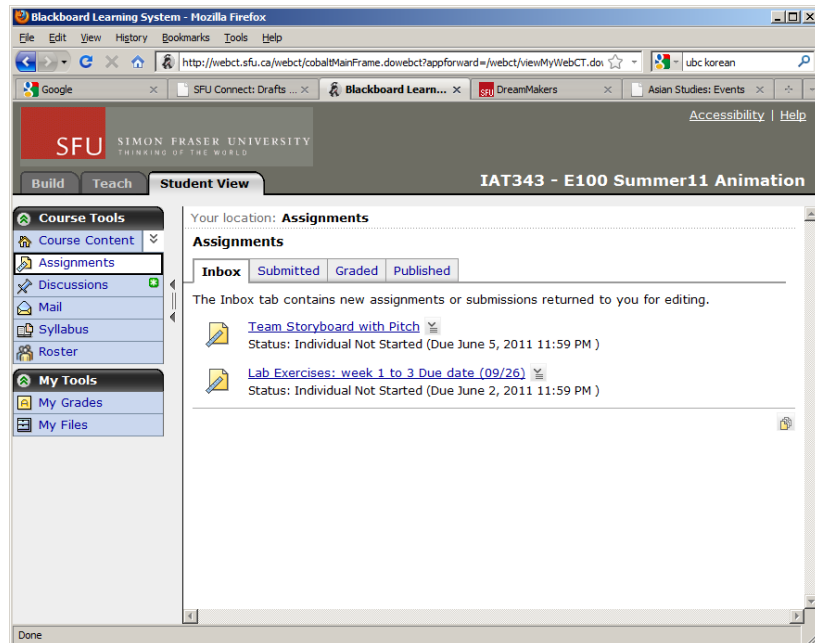
- First name:** Alexander
- Last name:** Akopyan
- E-mail:** aaa73@sfu.ca

The second profile shown is for a student with the following information:

- First name:** Ken

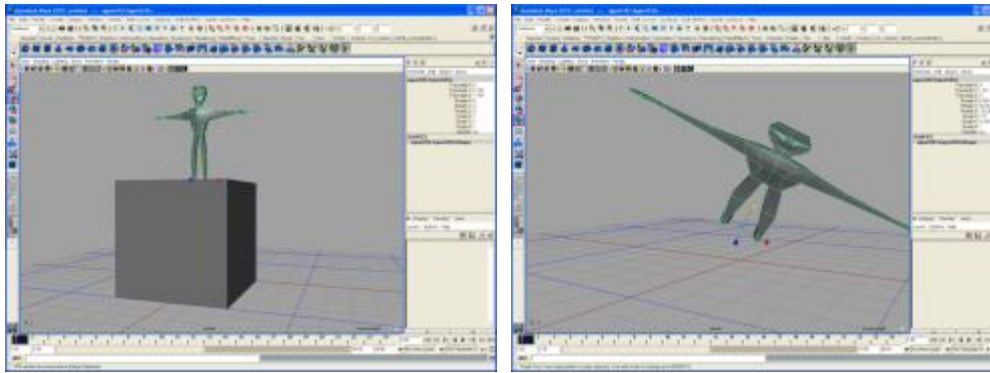
The browser's status bar at the bottom shows "Done".

WebCT > Course Tools > Assignments



- All the lab exercises in week 1, 2 and 3.
- Submit one Maya file ("LastName_Lab01.mb") per week by merging/importing all the exercises on each week.
- Design a simple website (no need to use Adobe Flash). <http://<your sfu webur>/iat343>
- Upload rendered images (minimum three with different camera angles) and all the files (e.g., Maya file (mb), texture) displayed on your website.
- Useful links:
 - SFU web page creation (http://www.sfu.ca/itservices/publishing/publish_howto/SFTPpublishing.html)
 - Automatic forwarding (http://www.htmlcodetutorial.com/document/index_tagstsupp_4.html)

Ex1. 3D Math 101: Transformation



$$P = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix}$$

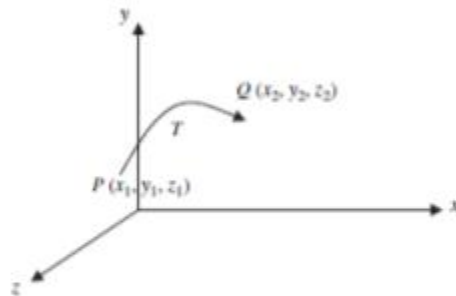


Fig. 4.3 Moving points in space

The transformation from P to Q is

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = T \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

Transformations

- Matrix Multiplication
- Linear transformations
- Scale transforms
- Rotation transforms

```
struct Vector3D
{
    float x, y, z;
};
```

```
struct Vector3D
{
    float pos[3];
}
```

$$\vec{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad 3 \times 1 \text{ matrix.}$$

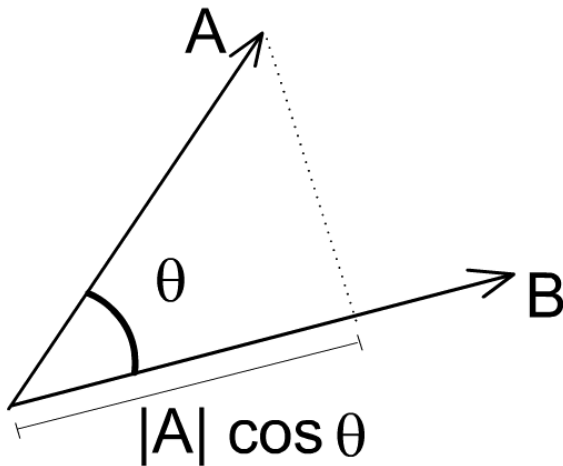
$$\begin{matrix} & \text{Column } j \\ \text{Row } i & \left(\begin{array}{c|c} & \\ \hline & e_i \\ \hline & \end{array} \right) \end{matrix}$$

Dot product

The dot product of two vectors $\mathbf{a} = [a_1, a_2, \dots, a_n]$ and $\mathbf{b} = [b_1, b_2, \dots, b_n]$ is defined as:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

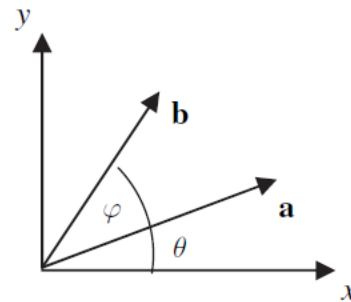
where Σ denotes **summation notation** and n is the dimension of the vector space.



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

where $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ denote the length of \mathbf{a} and \mathbf{b} and θ is the **angle** between them.

Why we need it?
Find angle between vectors.

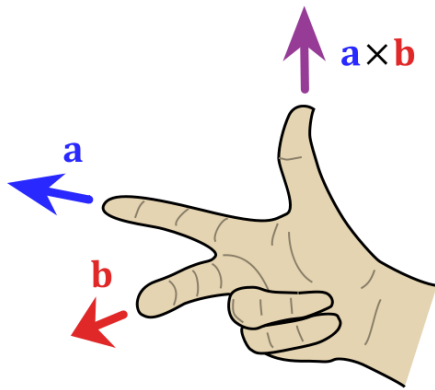
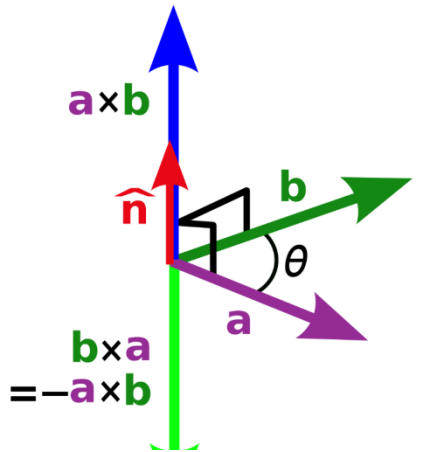


Source:
http://en.wikipedia.org/wiki/Dot_product

$$\begin{aligned} \mathbf{a} &= (|\mathbf{a}| \cos \theta, |\mathbf{a}| \sin \theta) \\ \mathbf{b} &= (|\mathbf{b}| \cos \varphi, |\mathbf{b}| \sin \varphi) \\ \mathbf{a} \cdot \mathbf{b} &= |\mathbf{a}| |\mathbf{b}| \cos \theta \cos \varphi + |\mathbf{a}| |\mathbf{b}| \sin \theta \sin \varphi \\ &= |\mathbf{a}| |\mathbf{b}| \cos(\varphi - \theta) \end{aligned}$$

Cross Product

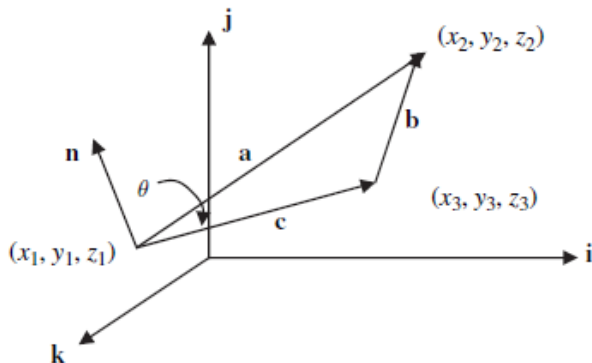
The cross product $\mathbf{a} \times \mathbf{b}$ is defined as a vector \mathbf{c} that is perpendicular to both \mathbf{a} and \mathbf{b} , with a direction given by the right-hand rule and a magnitude equal to the area of the parallelogram that the vectors span.



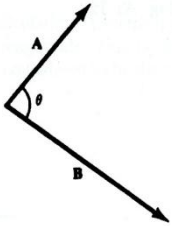
$$\mathbf{a} \times \mathbf{b} = ab \sin \theta \mathbf{n}$$

Source:

http://en.wikipedia.org/wiki/Cross_product



Dot Product & Normal Vector

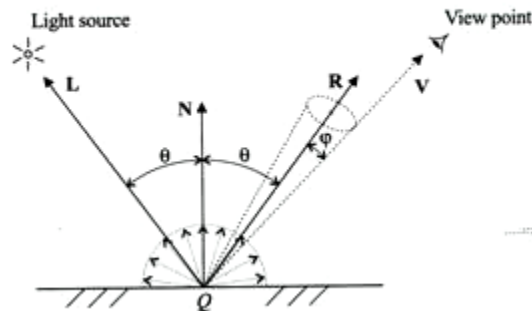
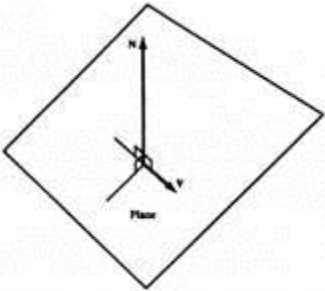


$$\mathbf{A} \cdot \mathbf{B} = |\mathbf{A}| |\mathbf{B}| \cos \theta$$

$$\cos \theta = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}| |\mathbf{B}|}$$

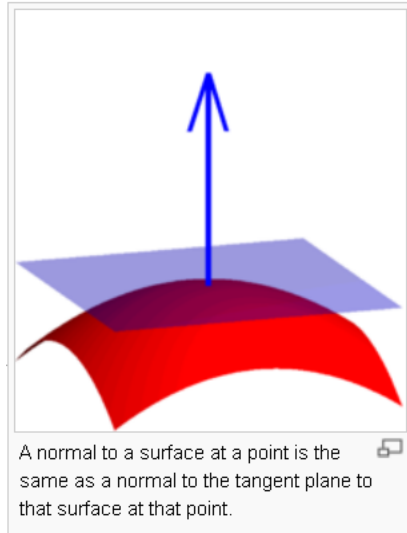
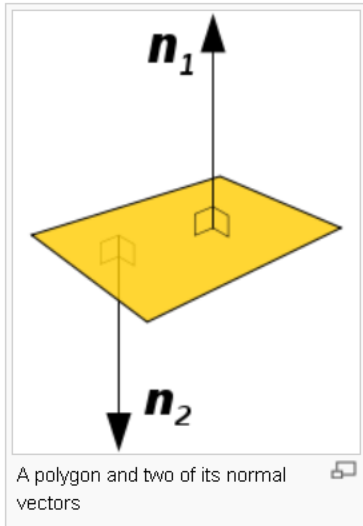
– Importance of Normal Vector

- Lighting simulation
- Polygon facing direction
- Dynamics & collision

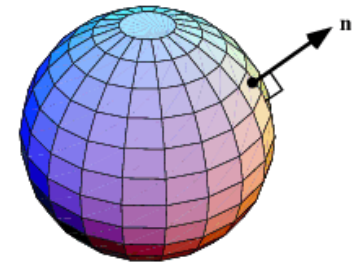
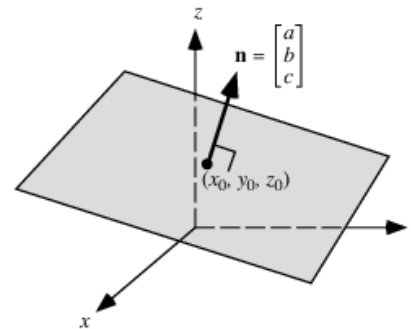


(Xiang & Plastock, 2001)

Surface Normal (Normal Vector)



A **surface normal**, or simply **normal**, to a flat surface is a vector that is perpendicular to that surface.



Source
http://en.wikipedia.org/wiki/Surface_normal

Source:
<http://mathworld.wolfram.com/NormalVector.html>

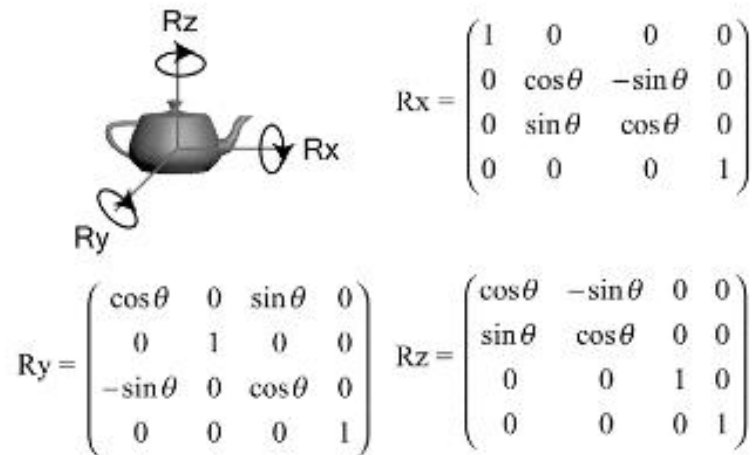
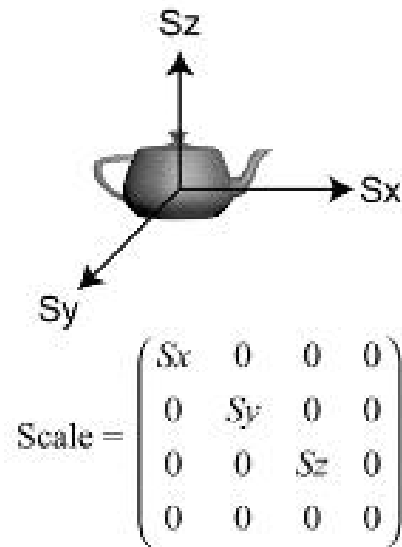
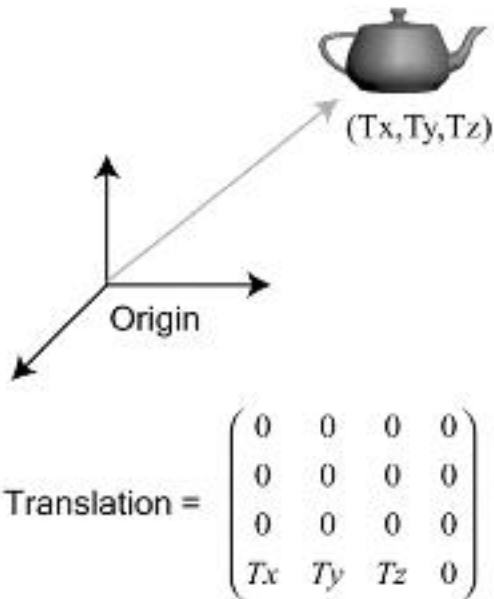
Matrix Algebra in 3D space

- Vector-Matrix multiplication
- Matrix-matrix multiplication & associativity
- Transpose of a matrix
- Identify matrix
- Inverse of matrix

Review Matrix algebra!

[http://en.wikipedia.org/wiki/Matrix_\(mathematics\)](http://en.wikipedia.org/wiki/Matrix_(mathematics))

Source: Introduction to 3D Game Programming with DirectX 9.0c: A Shader Approach by Frank D. Luna



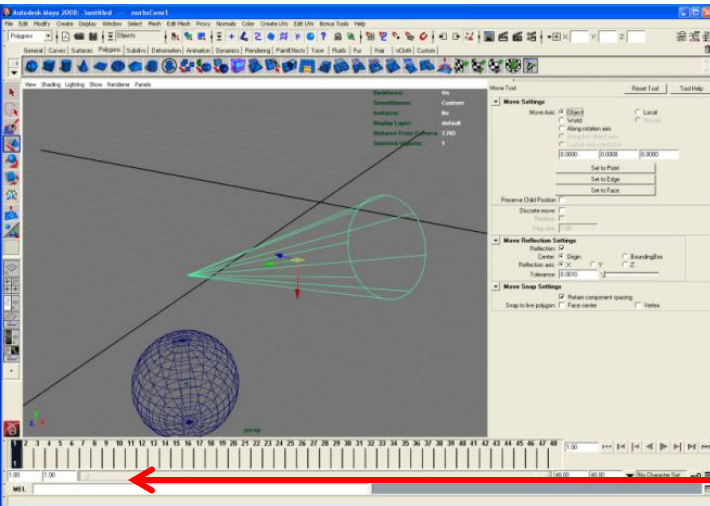
Source: Shaders for Game Programmers and Artists by St-Laurent, Sebastien

Using MEL (Maya Embedded Language)

User Guide > General > MEL and Expressions > Background > MEL Overview

- MEL™ is a scripting language at the heart of Maya. Maya's user interface is created using MEL, and MEL provides an easy way to extend the functionality of Maya. Everything you can do using Maya's graphical interface can be automated and extended using MEL. Familiarity with MEL can deepen your understanding of and expertise with Maya.
- You can take advantage of MEL without learning programming. For example, it's easy in Maya to perform some actions with the graphical interface, then drag the commands that resulted from the Script Editor to the shelf to create a button. However, learning MEL will open up new worlds to you, allowing you produce effects and save time in ways impossible using the graphical interface.
- Here are some examples of things you can do with MEL:
 - Bypass Maya's user interface, quickly create shortcuts, and access advanced features.
 - Customize Maya's interface and change defaults on a scene-by-scene basis.
 - Create procedures and scripts for custom modeling, animation, dynamics, and rendering tasks.

MEL (Maya Embedded Language)



- To create a cone
 - Type `cone` and press Enter (try with the numeric keypad).
- To transform the object.
 - `rotate 0 0 90 <Enter>`
 - `move 5 2 5 <Enter>`
 - `scale 2 1 1 <Enter>`
- To execute multiple commands.
 - `sphere; move 0 3 -3; scale 0.5 1 1 <Enter>`

sphere	-radius	5	-pivot	3 4 4	;
↑	↑	↑	↑	↑	↑
command	flag	argument	flag	argument	end of command character

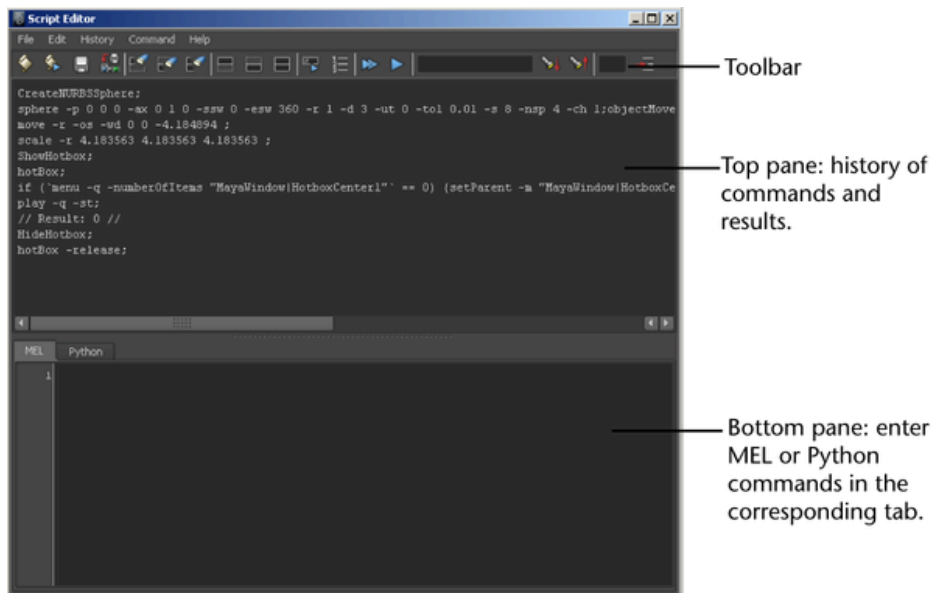
The above MEL command creates a NURBS sphere of radius 5 with its pivot at the origin and the sphere offset from the pivot in XYZ by 3 4 4.

Basic structure of Programming

- Data types and size
- How to declare variable
- Expressions: arithmetic, Boolean
- Simple input/output

Control Flow Statements

P:	Process	Input/output, break, continue
D:	Decision	if-else, switch-case,
I:	Iteration	for, while, do-while



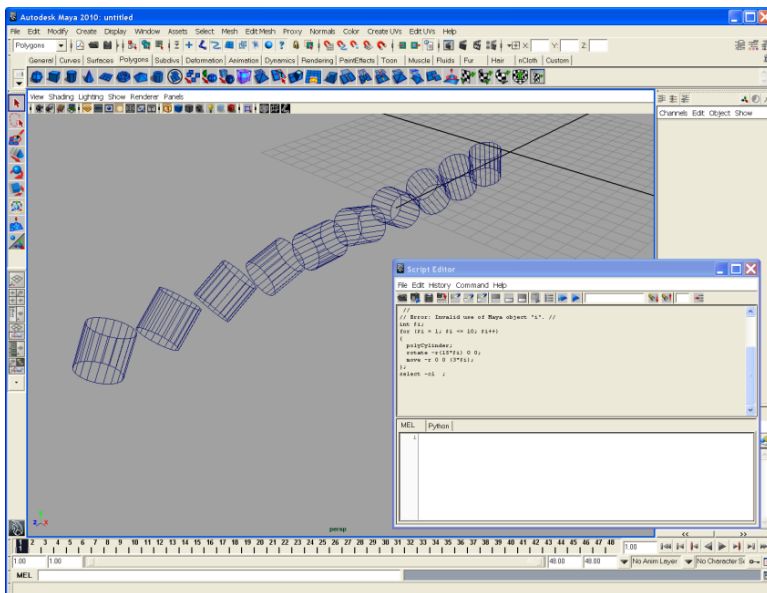
Generating multiple objects

1. Open Window > General Editors > Script Editor.
2. Using **for** loop: type simple statement;

```
int $i;
for ($i = 1; $i <= 10; $i++)
{
    polyCylinder;
    rotate -r(15*$i) 0 0;
    move -r 0 0 (3*$i);
};
```

To execute the script do any of the following:

- Press the Enter key on the numeric keypad
- Select Command > Execute.
- Select the text you want to execute and press Ctrl + Enter



1. Create two variables representing vector, add them and assign it to new vector;

```
vector $a = <<1, 0, 0>>;
```

```
vector $b = <<0, 1, 0>>;
```

```
vector $c = $a + $b;
```

2. Go to **Command > Execute** to see the result of vector addition.

3. To find the length of vector 'c';

```
float $c_length = mag($c); // execute
```

4. To find the angle between vector 'a' and 'b';

```
float $ab_angle = angle($a, $b); //execute
```

5. To convert to angle, not radians;

```
$pi = 3.1415;
```

```
float $ab_angle_degrees = $ab_angle*180/$pi;
```

6. To generate the cross vector;

```
vector $c = cross($a, $b);
```

7. To apply the dot operator;

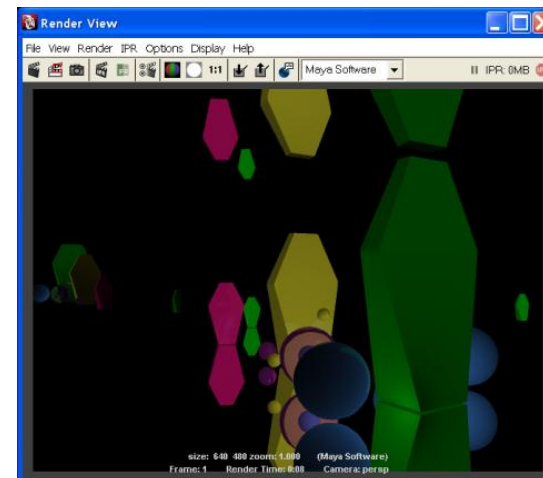
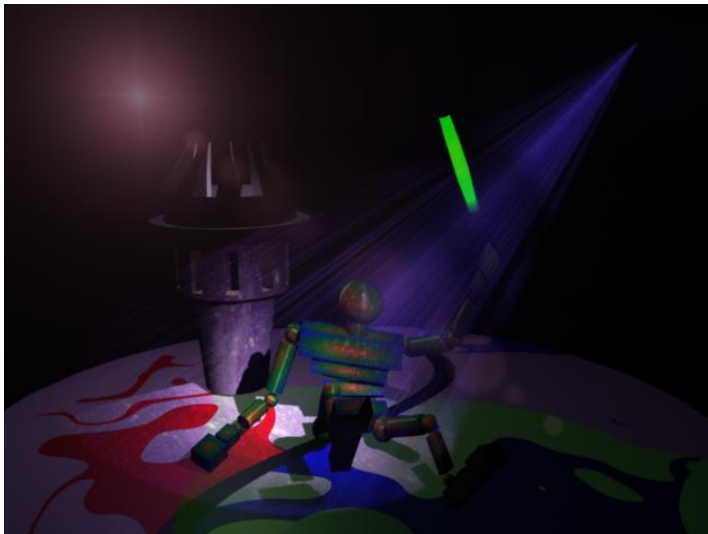
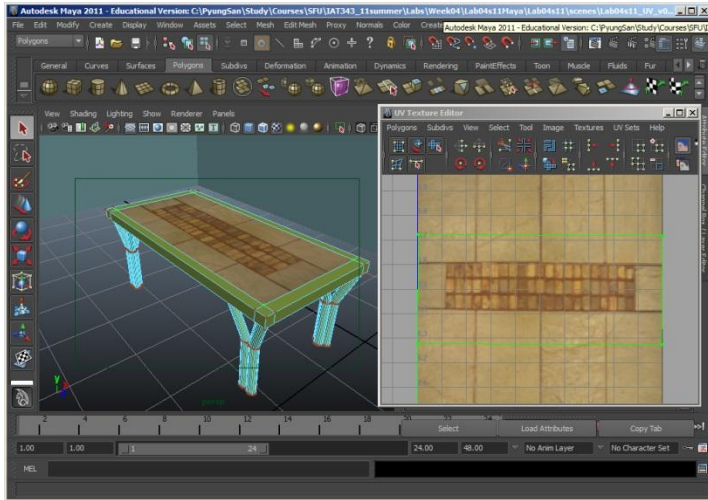
```
$ab_dot = dot($a, $b);
```

8. Define a new vector parallel to \$a and generate a dot operation between new vector and \$a to prove it. What's the result? Save your file (File > save script 'Lastname_3dMatrix.mel')

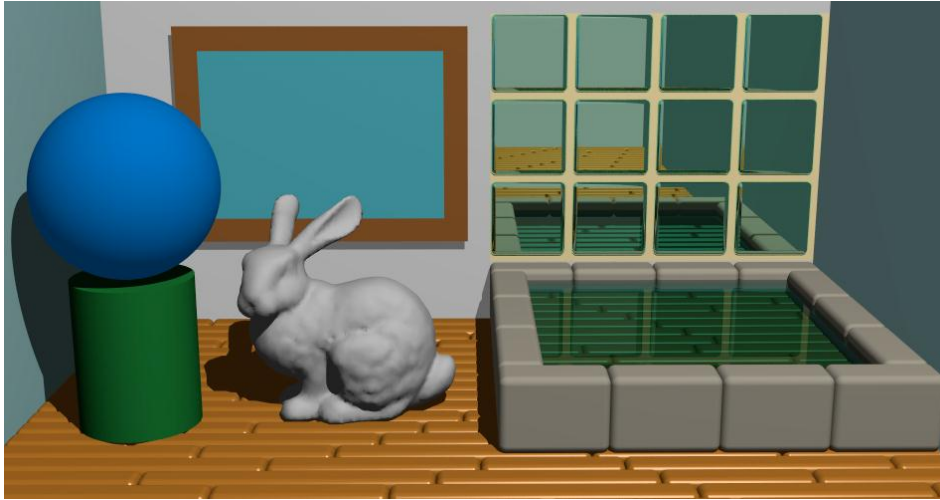
Exercise 1.

Today's Lab

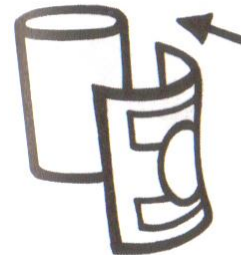
- UV maps
 - Creating Photoshop network
- Reflection
- Lighting



Texture Mapping



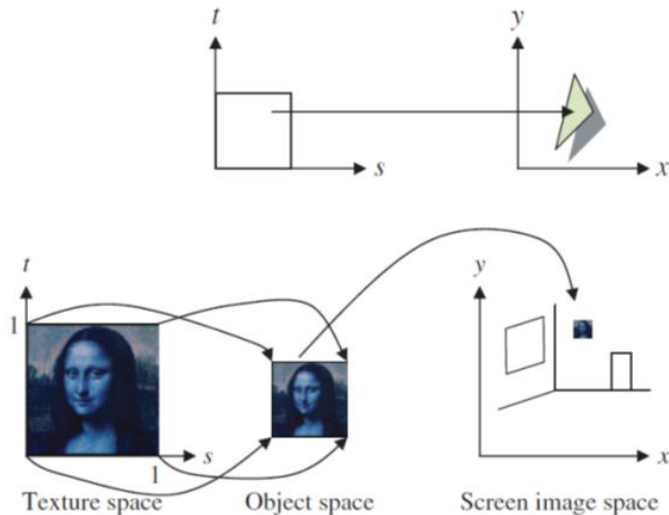
- Wrap image onto 3D geometry
- Applying raster image information to CG geometry
- Add a great detail to enhance realism
- Avoid the need for massive amounts of geometric detail



•Source:
(Watt, 1999), (Masson, 2004) & (Park, 2004)

Texel(Texture elements):
Array of pixels

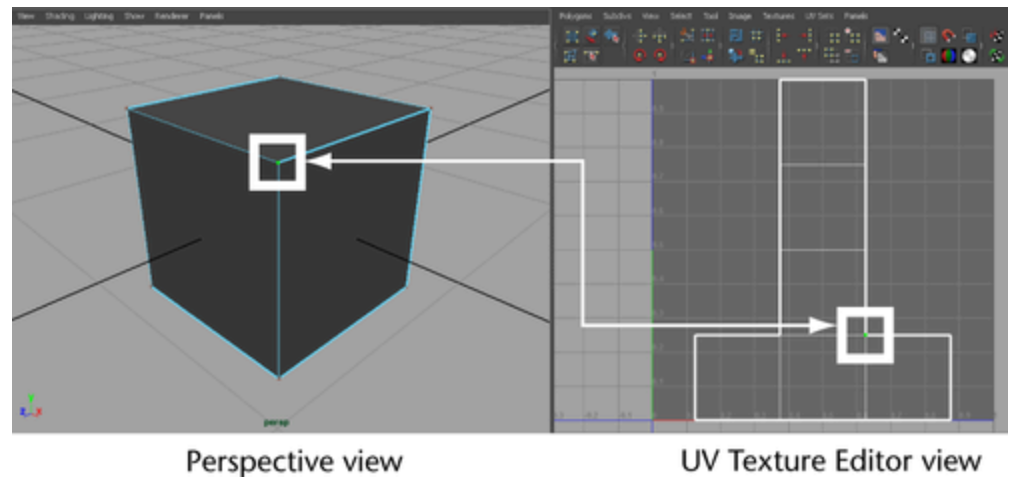
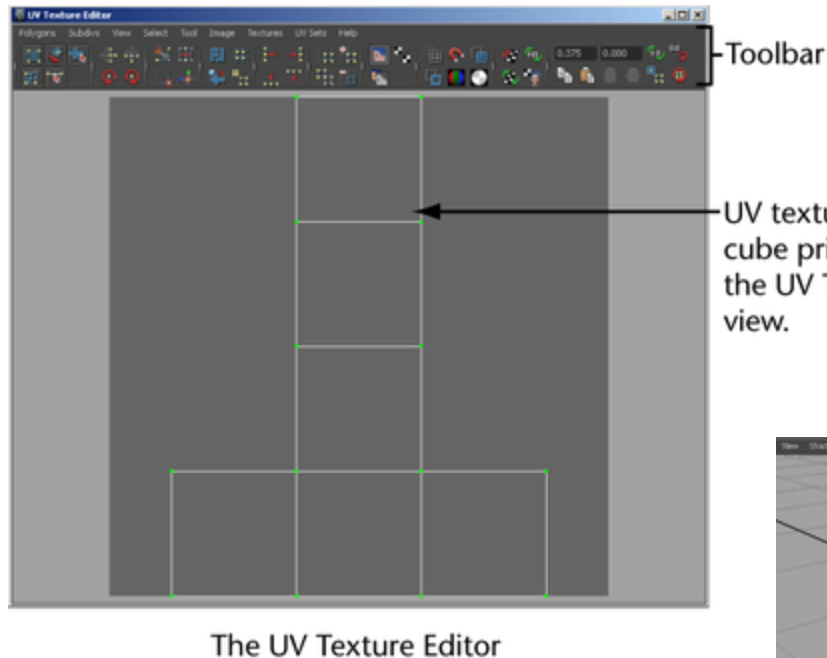
Mapping between
(s,t)-texel space and
(x, y, z)-object surface space



UVs (pronounced U-VEEZ) are two-dimensional texture coordinates that reside with the vertex component information for polygonal and subdivision surface meshes. UVs exist to define a two-dimensional texture coordinate system, called **UV texture space**. UV texture space uses the letters U and V to indicate the axes in 2D. UV texture space facilitates the placement of image texture maps on a 3D surface.

The process of creating explicit UVs for a surface mesh is called **UV mapping**. UV mapping is a process whereby you create, edit, and otherwise arrange the UVs (that appear as a flattened, two-dimensional representation of the surface mesh, over top of the two-dimensional image to be used as a texture as it appears in the UV Texture Editor).

User Guide > Modeling > Mapping UVs > UV mapping overview > UV Texture Editor overview



Selecting a UV component in either the 3D scene view or the 2D UV Texture Editor view displays the selected UV in both views

Shaders

Lambert Material

This material type is the most basic and does not include any attributes for specular. This makes it perfect for matte surfaces that do not reflect the surrounding environment. The Lambert material type can be transparent and will refract in a Raytrace rendering, but without any specular, it won't reflect.



Phong Material

This material adds a sharp specular highlight to the Lambert material. The size and intensity of the highlights are controlled by the Cosine Power attribute. This material can also have reflections from either an environment map or Raytraced reflections. The Phong material is good for plastics.



PhongE Material

This material type adds a different kind of specular highlight to the Lambert. The PhongE material includes attributes such as Roughness that controls the softness of the highlight, Whiteness that controls its intensity and Highlight Size.



Blinn Material

Many artists use this material type exclusively because it offers high-quality specular highlights using attributes such as Eccentricity and Specular Roll Off. This material type can be edited to look like a Phong material, which has sharper highlights, in cases where you need better anti-aliasing of highlights during an animation. This material is good for glass and metals.



Anisotropic Material

This material type simulates surfaces which have micro-facet grooves and the specular highlight tends to be perpendicular to the direction of the grooves. Materials such as hair, satin and CDs all have anisotropic highlights.



Shading Map Material

This material type allows you to create custom shading on surfaces. A ramp texture controls the positioning and color of the shading and highlights on the surface. If you want to emphasize the dark areas, simply darken the lower end of the ramp.



Ramp Shader

This shader gives you extra control over the way color changes with light and the view angle. You can simulate a variety of exotic materials and tweak traditional shading in subtle ways. All the color-related attributes in the Ramp Shader are controlled by ramps.



Ocean Shader

The Ocean shader is a specialized shader with attributes defining realistic waves on large bodies of water. It is usually used through the Fluid Effects > Ocean > Create Ocean command, which automatically creates nodes required to render an ocean.



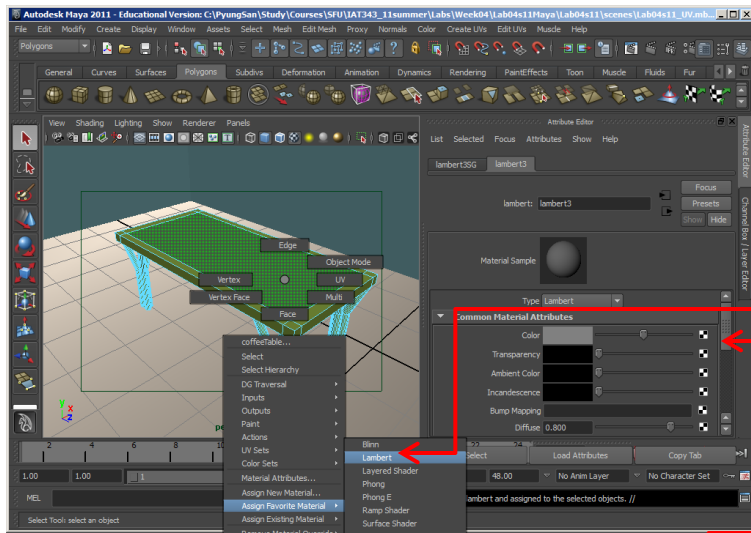
Hair Tube Shader

Hair tube shader simulates a thin tube, where the width of the tube is small enough that local shading effects can be ignored. All shading derives from the view and the tube direction. Because the highlights are spread across the entire tube width, rendering fine hairs does not require as high anti-aliasing levels.



(Autodesk, 2007)

Ex 3.1 UV Edit

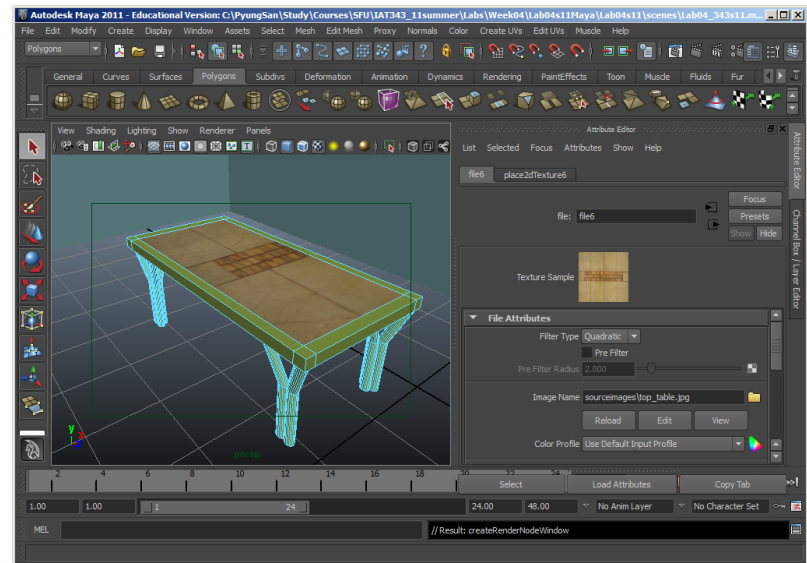
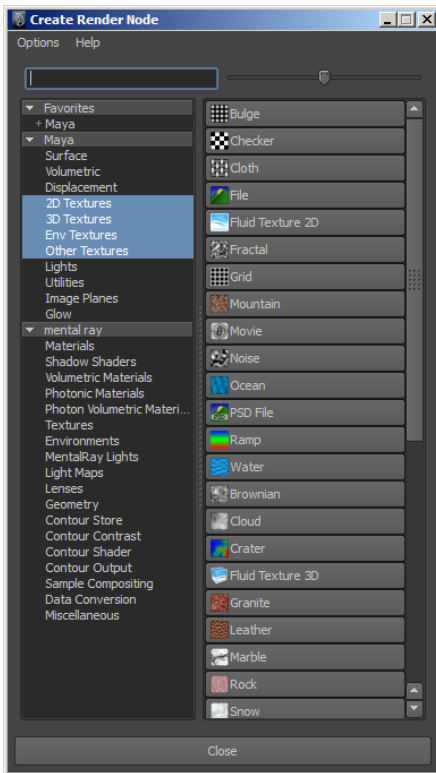


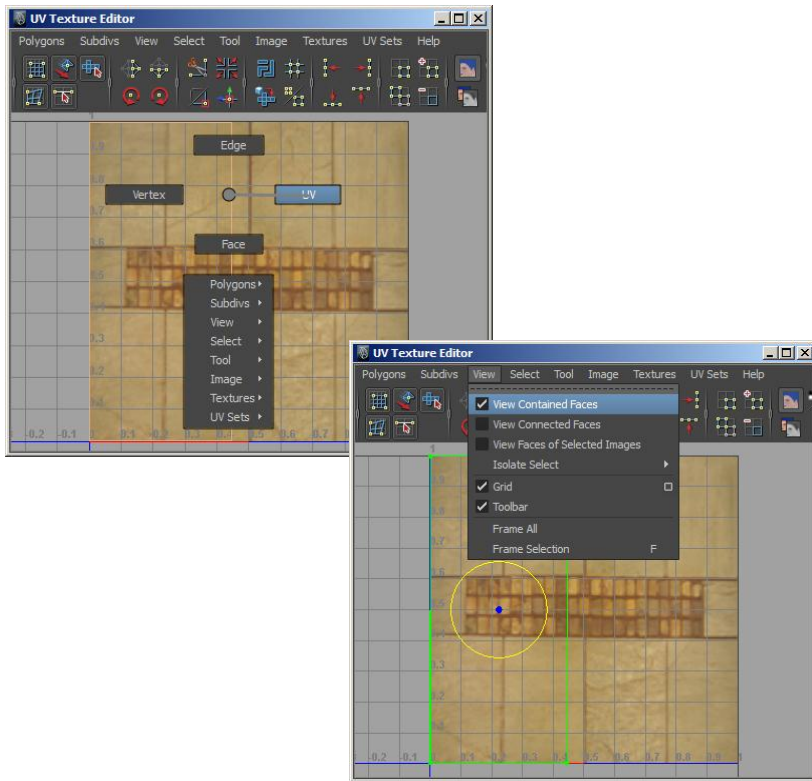
1. Open 'Lab04s11_UV.mb' file. To apply an image on the top of the table, first select the table and focus zoom on the target object (press 'F'). Press '6' to activate hardware texturing preview.

2. Select one top face of the table and apply Lambert material while holding down Right Mouse Button (**Assign Favorite Material > Lambert**).

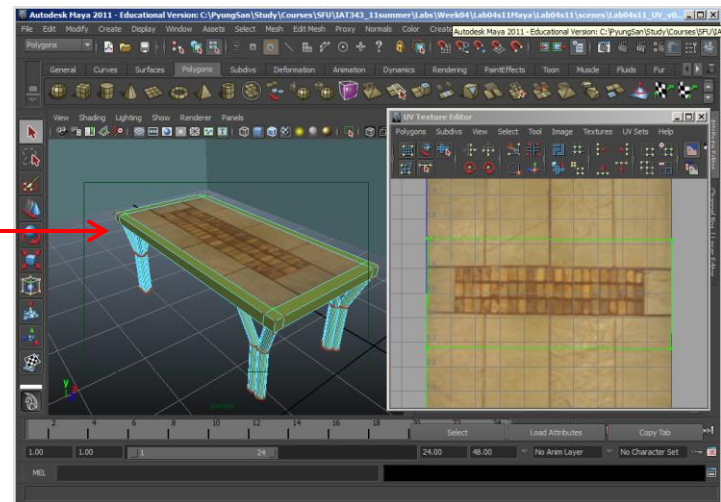
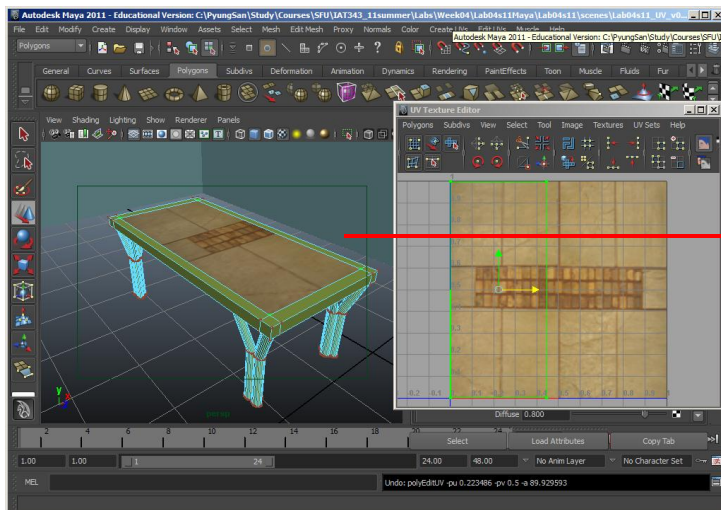
3. To apply an image on the selected face, clicking on the map icon (checker pattern) next to **Color** parameter, which opens **Create Render Node UI**.

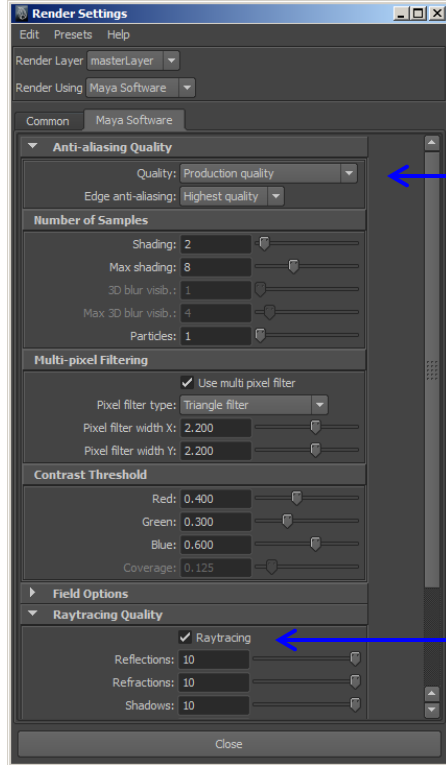
4. Click on **File**, look for **Image Name** parameter and click on the folder icon. Select an image ('top_table.jpg') under **sourceimages** subfolder.





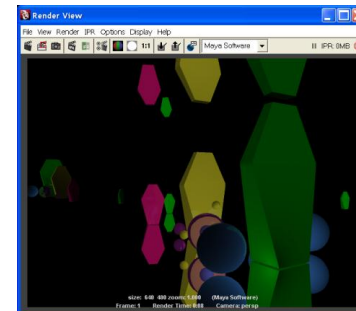
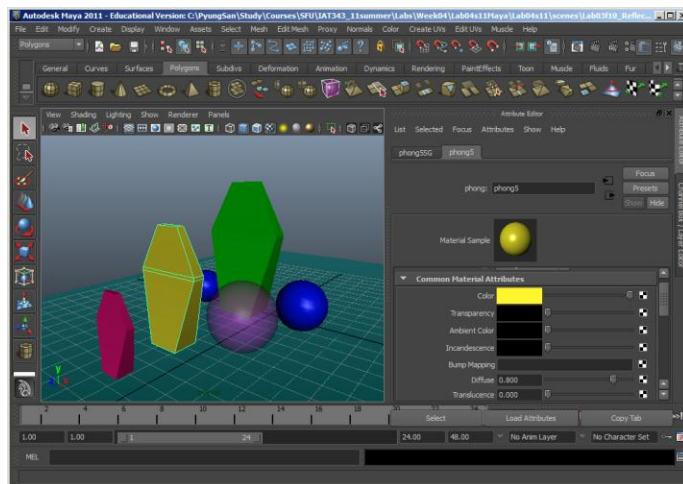
5. Currently, the image is not aligned properly. To fix it, select the target face and open **UV Texture Editor** (Windows >). Under View menu, check **View Contained Faces** option.
6. Place your mouse cursor on the **UV Texture Editor** and press/hold RMB to select the **UV**. Drag the mouse cursor to select all the UV points of the face.
7. Using the regular transformation tool to adjust the mapping direction, we can translate, rotate or scale the UV space.
8. To see the results, hit the **Render the current frame** button.
9. Save your file.



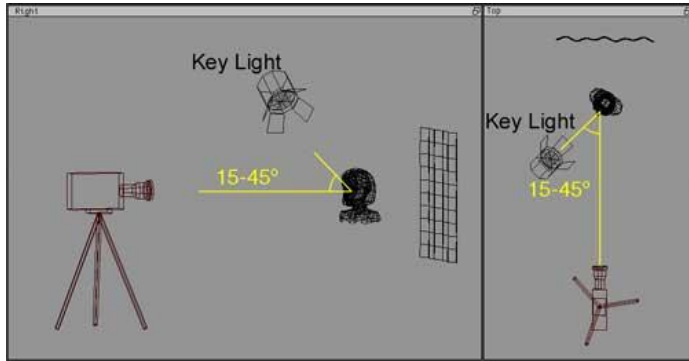


Exercise 3.2: Creating Reflection

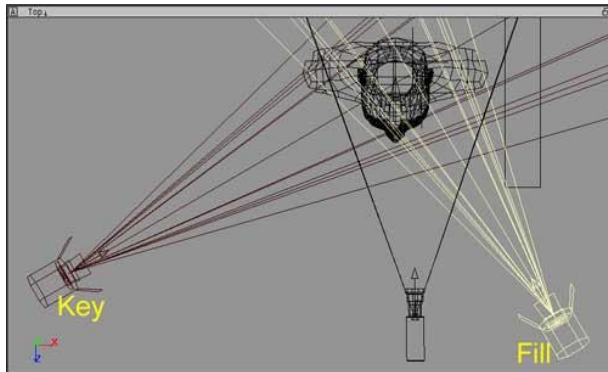
1. Create a couple of primitives including the Polygon Plane. Assign different shaders which has the reflection parameter (e.g., Phong).
2. Open the **Render Settings** window (**Window > Rendering Editors > Render Settings**).
3. Switch to the **Maya Software** tab. To improve rendering quality, select Production Quality of Quality parameter.
4. Click on the triangle next to the **Raytracing Quality**. Click on the checkbox: **Raytracing**. Play with different values in the **Reflection**. To see a quick result, hit the **Render the current frame** button.
5. To generate refraction of the target object, select the object first, and open the material shader section of the object. (Ctrl + A). Modify **Transparency** value to high (drag the slider to the right), and turn on the **Refractions** option in the **Raytrace Options**. Play with the **Refractive Index** value to see different results.



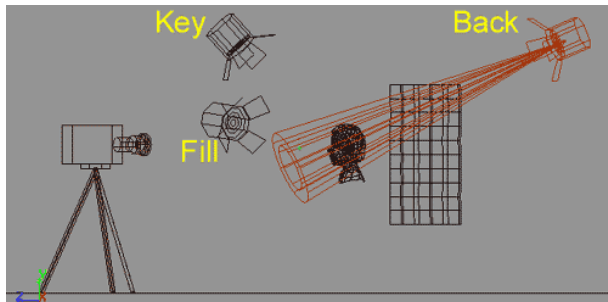
Three Point Lighting



Key Light



Fill Light



Back Light



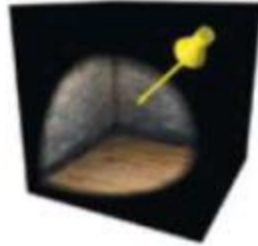
Reading material: Jeremy Birn
<http://www.3drender.com/light/3point.html>

Light Types in Computer Graphics



Spot

Spot lights emit light that radiates from a point within a limited cone angle. You can use this cone angle to limit the area receiving light.



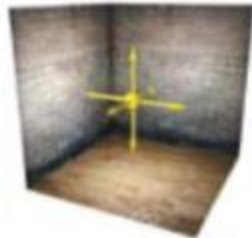
Directional

Directional lights use parallel rays of light to illuminate a scene. Shading is very uniform without any hotspots. These rays are similar to the light of the sun, which hits the earth with parallel rays.



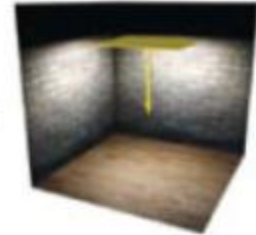
Point

Point lights emit light in all directions, radiating from a single point. This creates an effect similar to a light bulb. This light creates subtle shading effects with definite hot spots.



Area

Area lights emit light using a two-dimensional area. The area light's icon can be used to help define the light's direction and intensity. A larger area light has a stronger intensity.



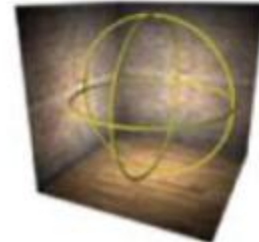
Ambient

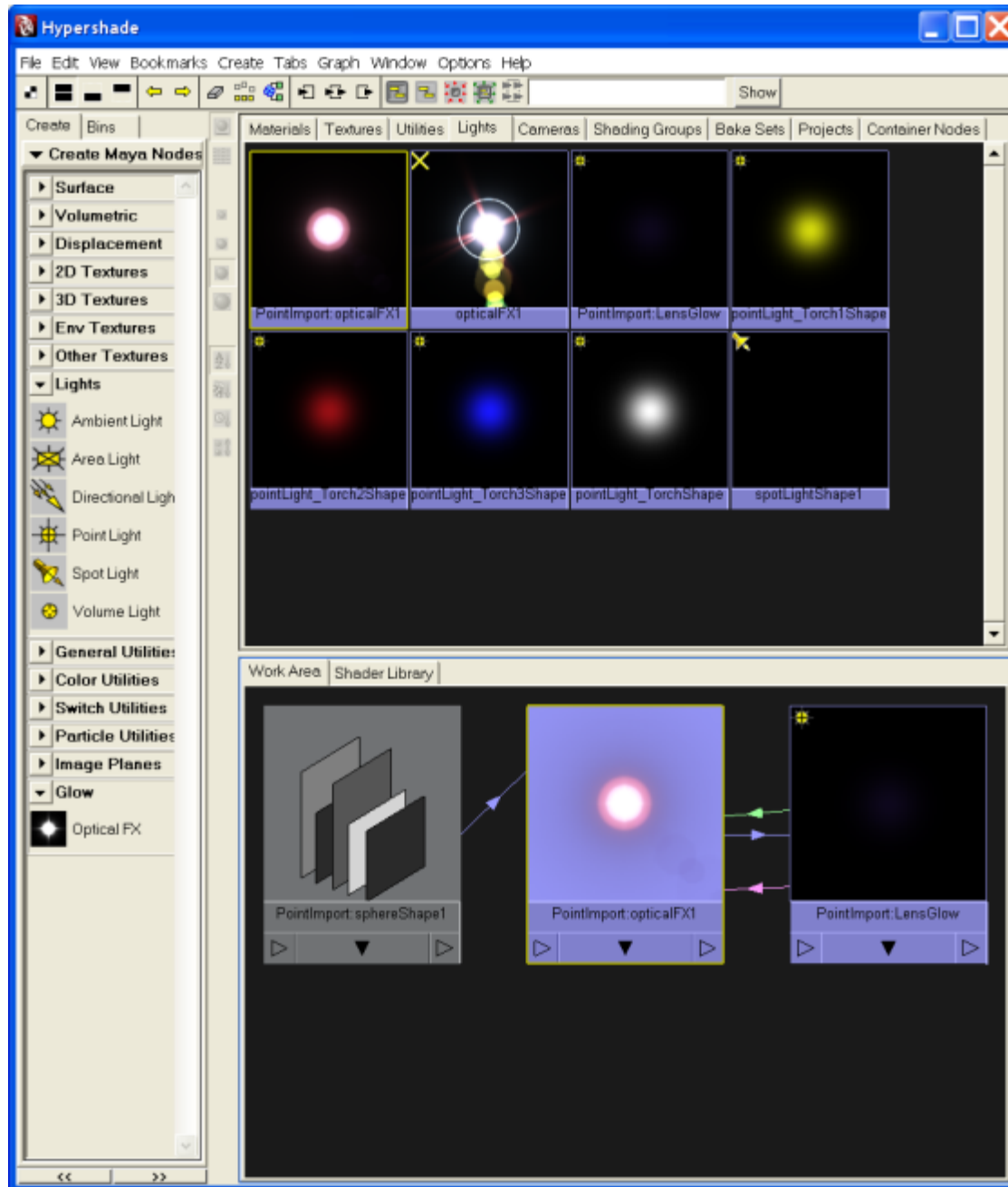
Ambient lights emit light uniformly in all directions. The **Ambient Shade** attribute adds positional behavior. Bump maps are not visible with ambient light alone.



Volume

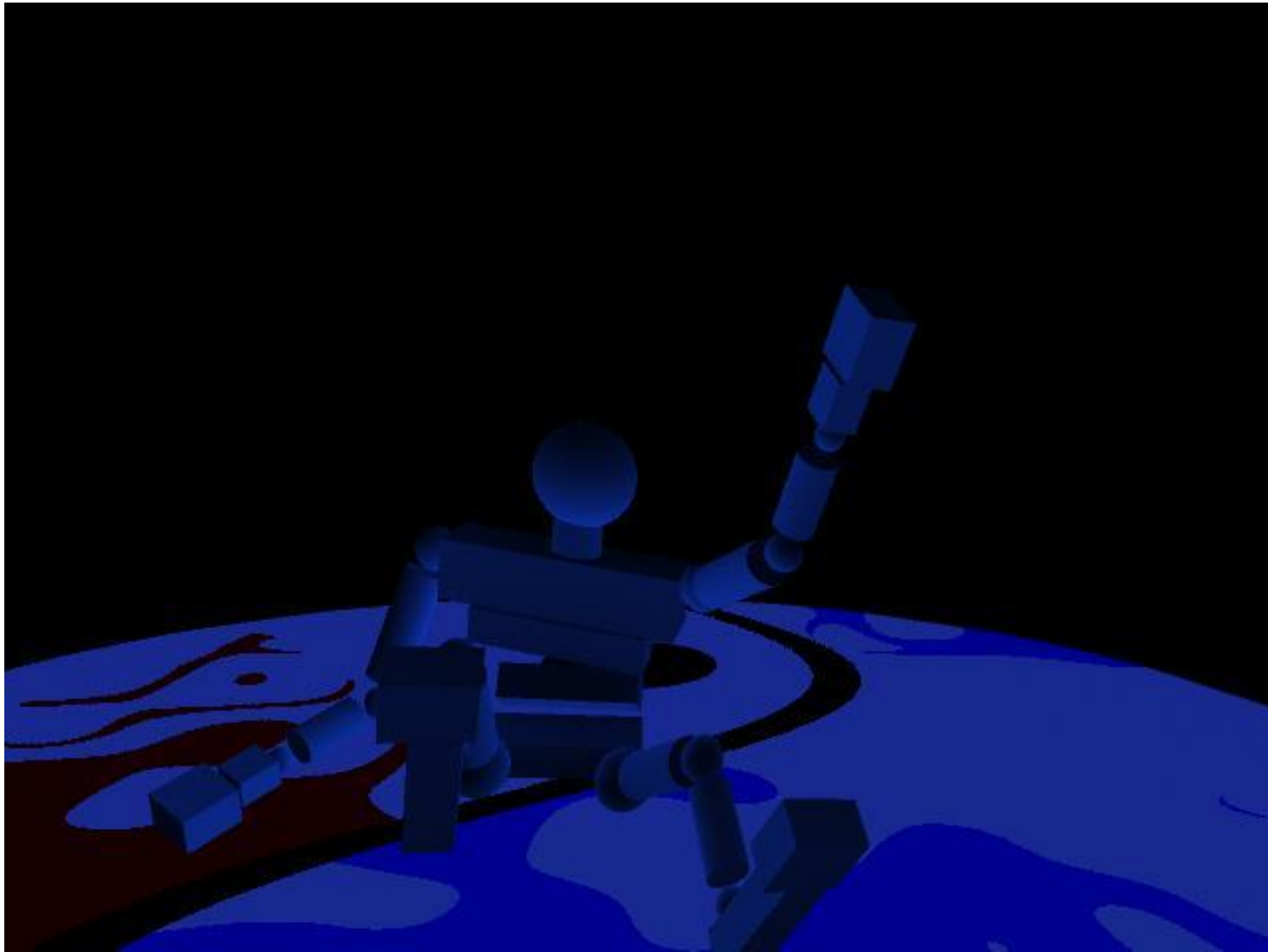
Volume lights emit light in all directions for a finite distance based on a 3D geometric shape. The light shape can be a box, a sphere, a cylinder or a cone.



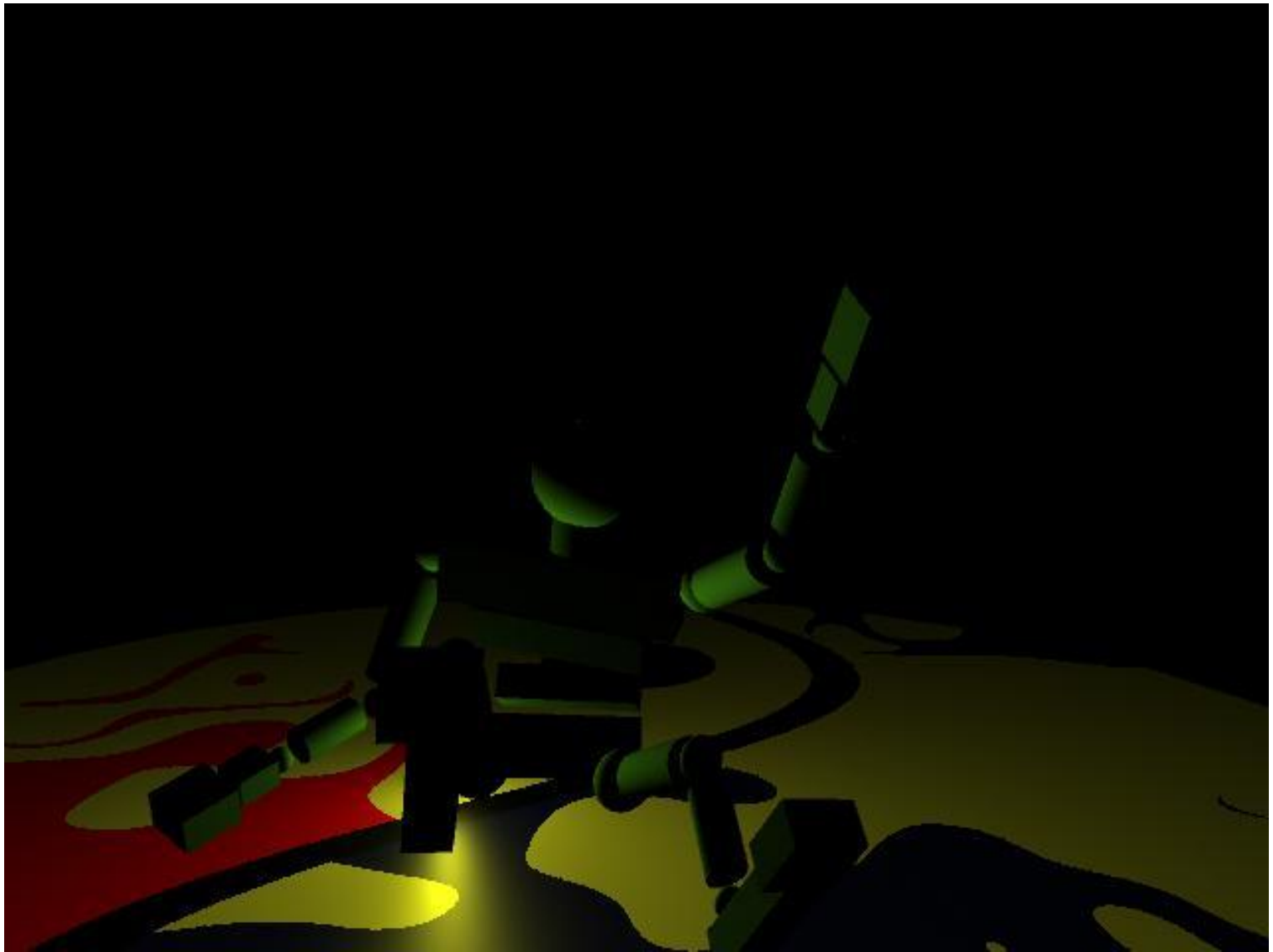


Lights in Maya

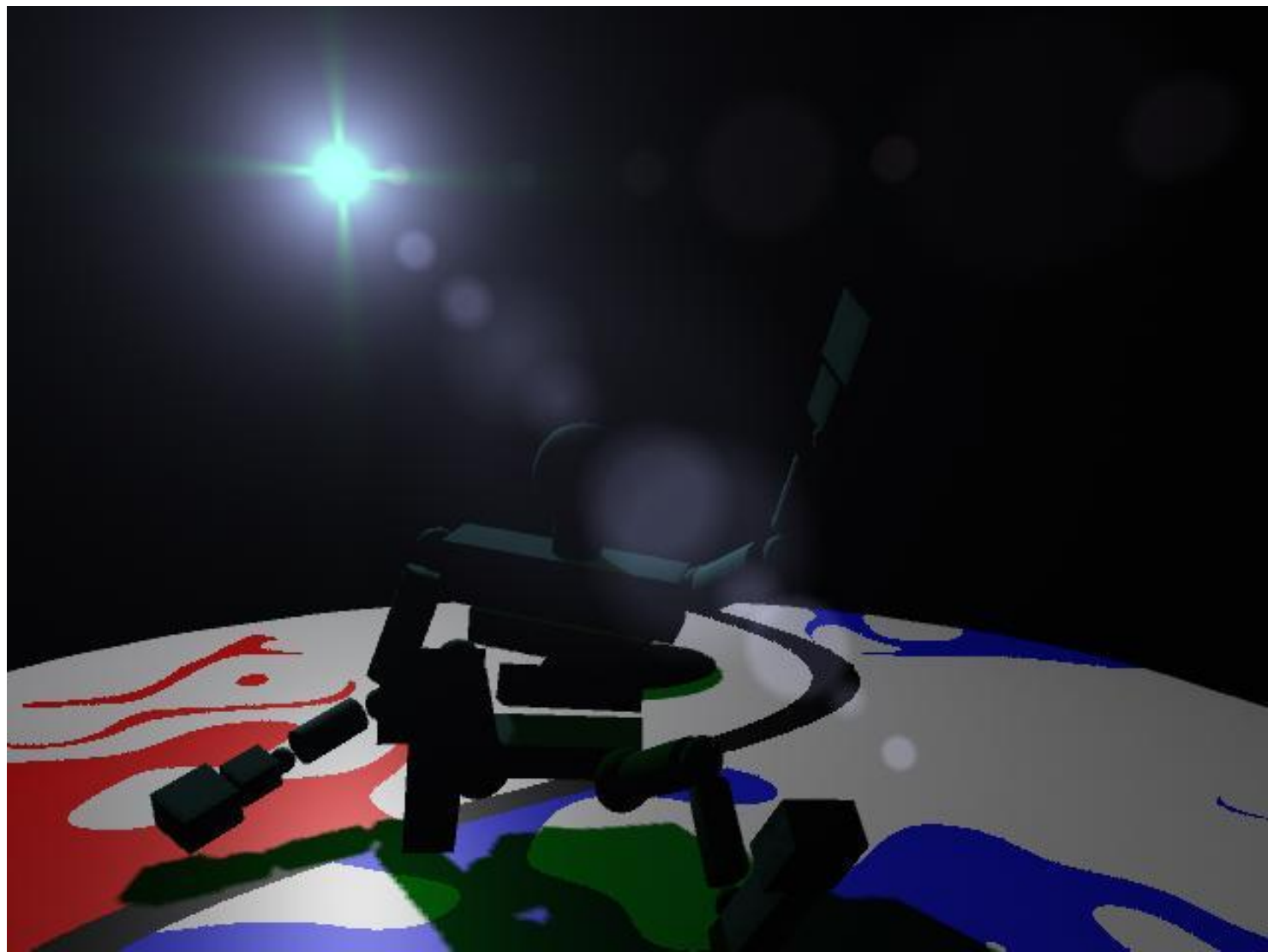
- Ambient light
- Area light
- Spot light
- Directional light
- Volume light
- Point light



Ambient light



Spot light



Spot light with lens flare

Spot Light Attributes

Intensity

This attribute determines how much light is emitted from the light source. As you increase the Decay and Dropoff values, you need a more intense light.

Decay

This attribute determines how much the light intensity diminishes as the light gets further from its source. Therefore, if you choose to use Decay, you need to increase the Intensity.

Cone Angle

This attribute determines the width of the spot light's cone of influence. The areas outside the cone are not illuminated.

Color

You can set RGB values for the light being emitted. This will have an influence on the color of your scene.

Hotspot

The point where the light is most intense is referred to as the hotspot. You also know it as a specular highlight. The look of the highlight is a result of the intensity of the light and the shading qualities of the surface's Material node.

Dropoff

This attribute determines how much the light intensity diminishes as it gets to the outer edge of the light. This puts more emphasis on the light's hotspot.

Penumbra Angle

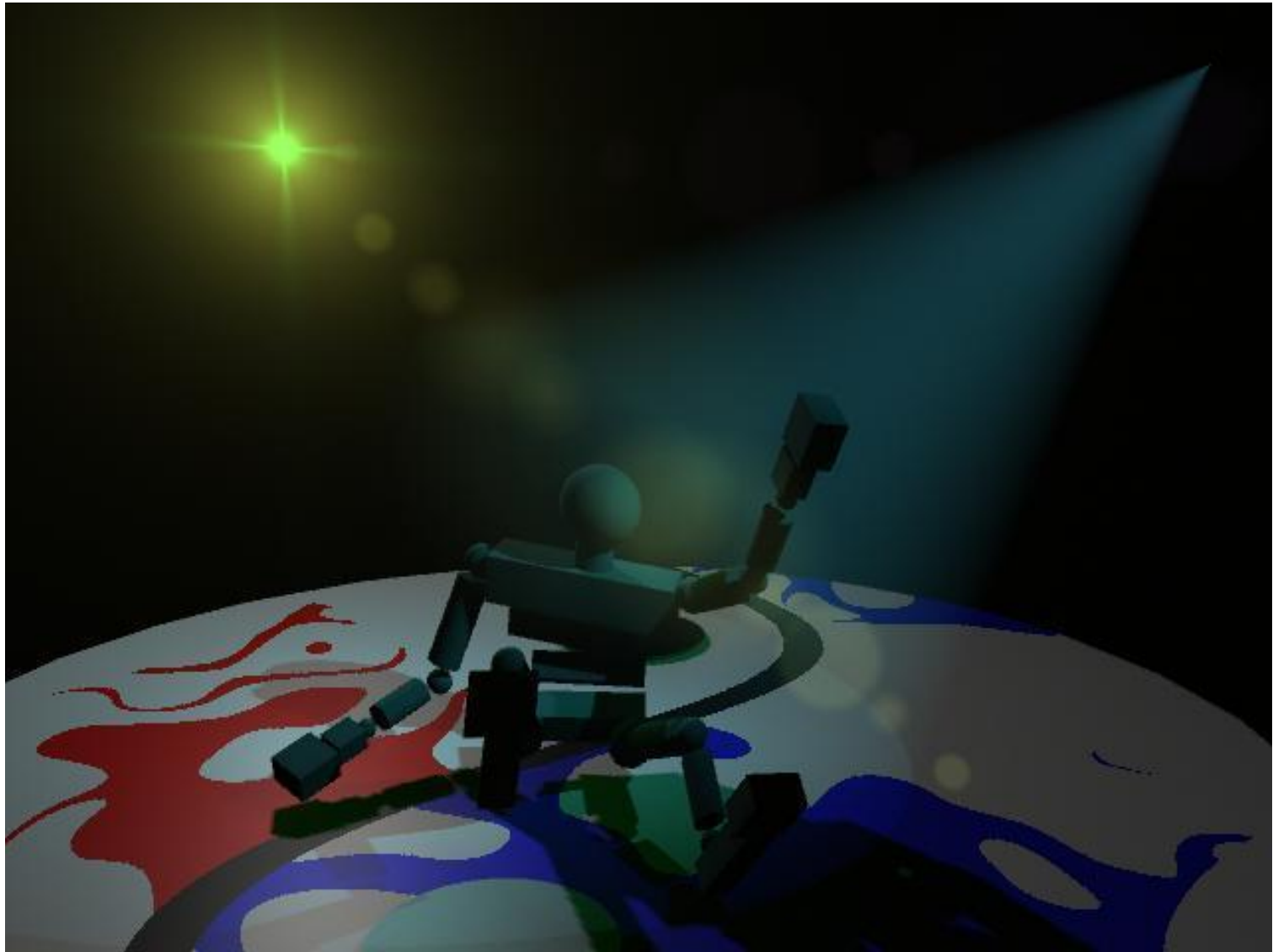
This attribute creates an area at the edge of the spot light where the light fades. A larger value here creates a soft look for the light.



Decay Rate



- No Decay : Light reaches everything
- Linear: Light intensity decreases proportion to distance ($1/d$)
- Quadratic: Light simulation close to real life ($1/d^2$)
- Cubic: Faster decay rate than in real life ($1/d^3$)

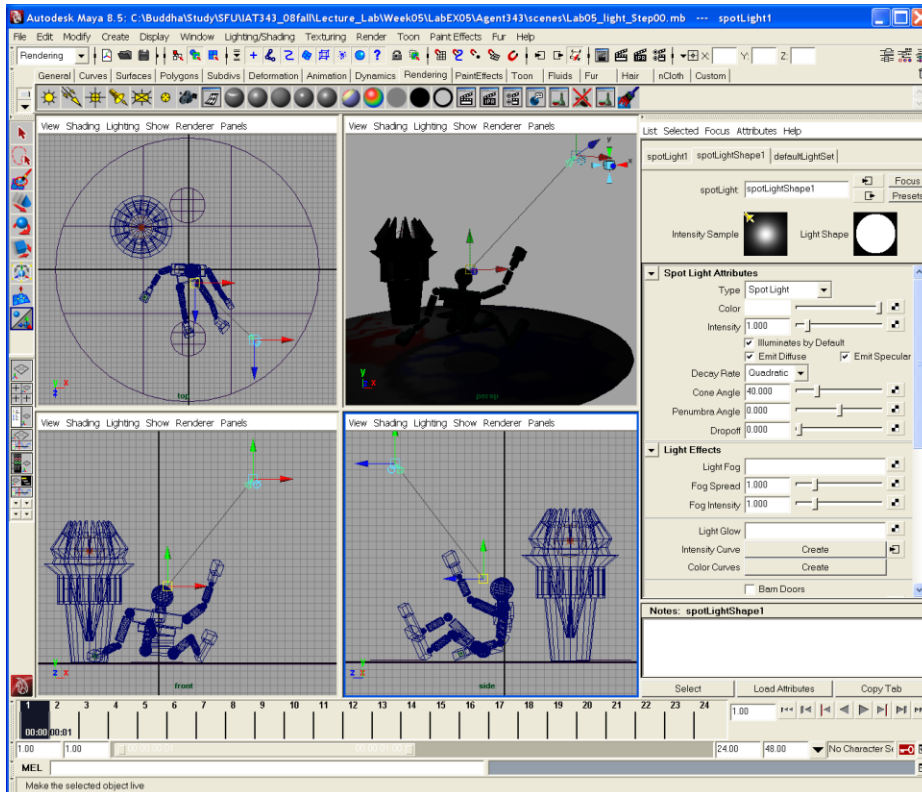




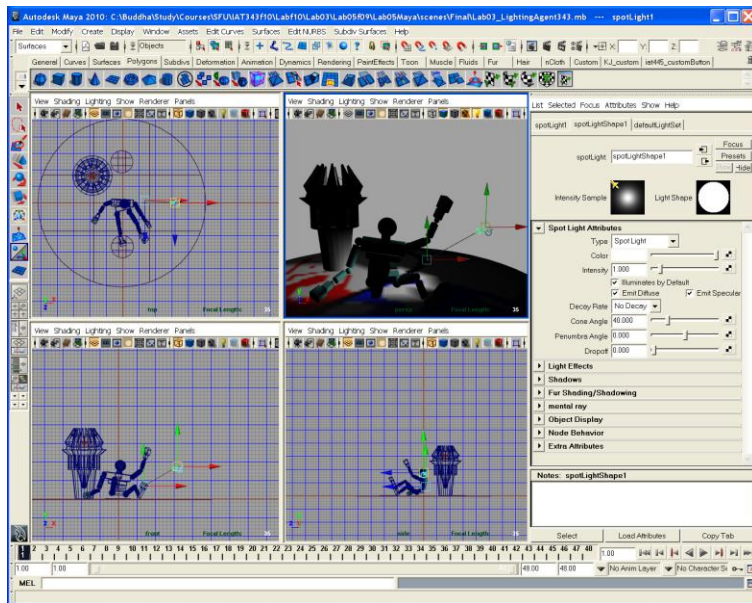
Texture mapping on Color attribute

Exercise 3.3: Lighting

Spot Light

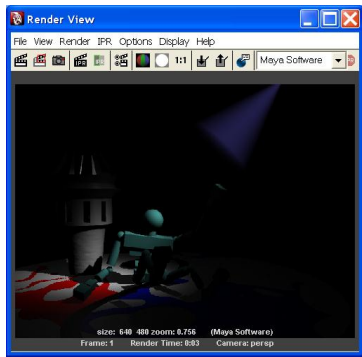
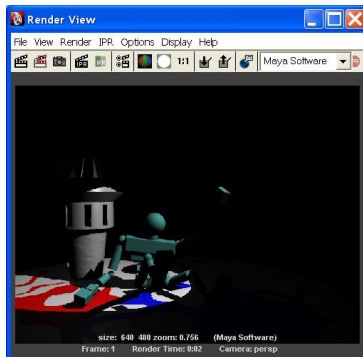
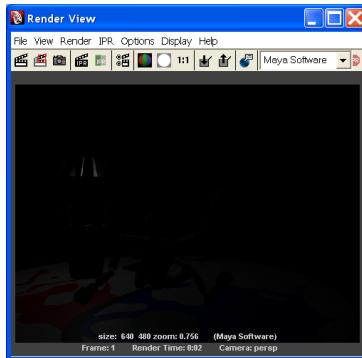
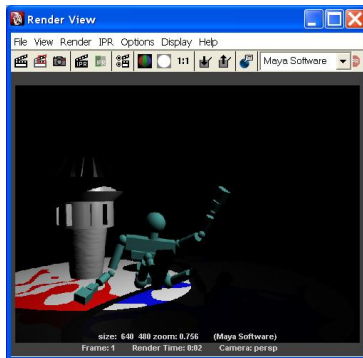


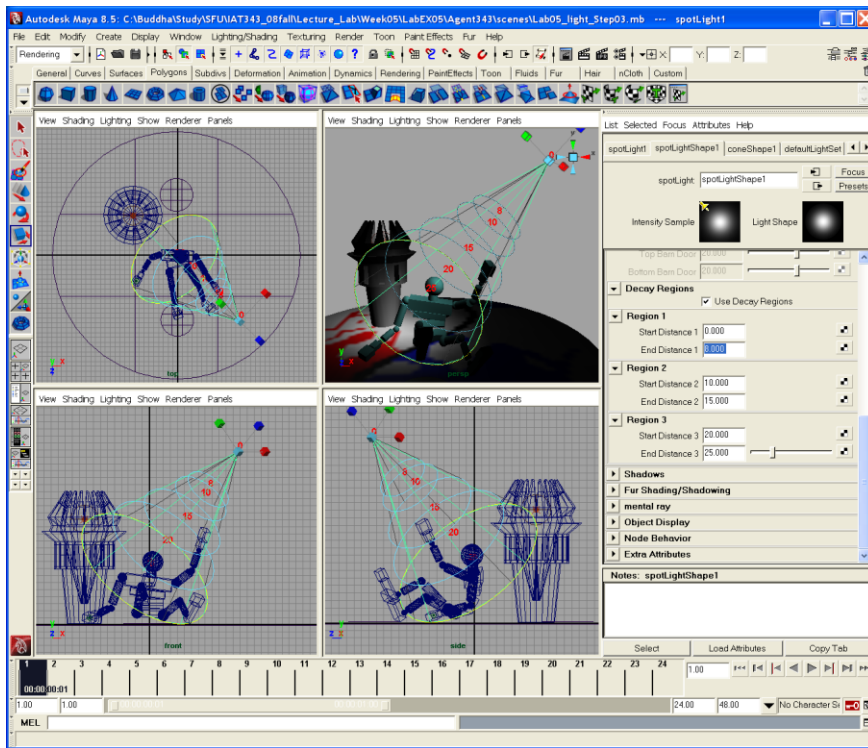
1. Open **Lab04_LightingAgent.mb** file.
2. This scene has only one light, **pointLight_Torch**. To see the light effect in the Perspective viewport, click on the target view and press **7**.
3. First, let's create a spot light. Select the **Create > Lights > Spot Light > Option Box**. To create the spot light at the center of the grid, uncheck **Interactive Placement** option. Click **Create** button. You can see the spot light icon (a cone with an arrow) .
4. To place the spot light pointing down the character, use different navigation tools (**View > Camera Tools**) in the viewport to achieve precise control over camera position as well as apply different transformation tools (Move) to adjust the position, rotation and the direction of the **Spot Light**.
5. Besides using the standard TRS (translation, rotation, scaling) tool, the Manipulator tool (**Modify > Transformation Tools > Show Manipulator Tool**) brings two manipulators (**Look-at point & Eye point**) to position and to aim the light accurately. If it is necessary, adjust the positions of both manipulators by click/drag.



Decay Rates, Shadow & Color

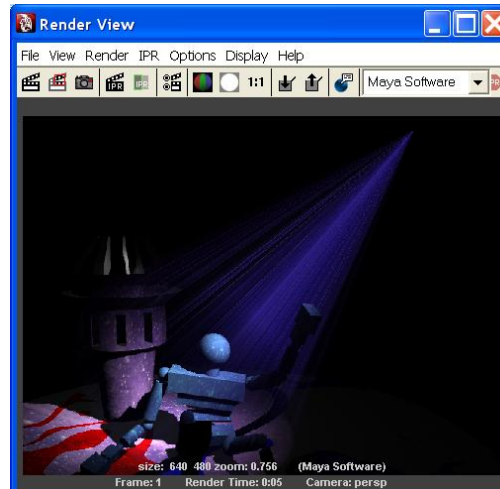
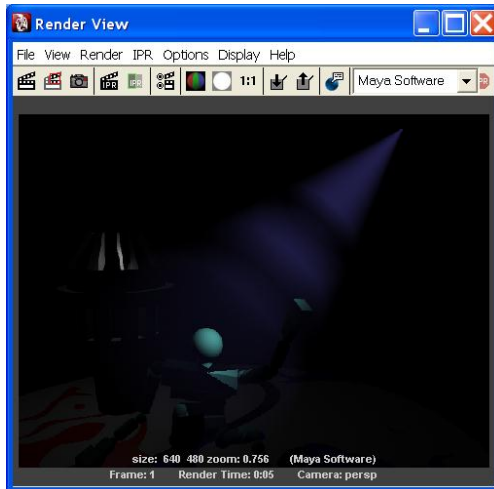
5. Depending on how light diminishes with distance (decay rate), it brings different mood as well as visual information. In the **Spot Light Attributes** section (Ctrl + A to open it), there are four settings for controlling **Decay Rate**. Select **No Decay** or **Quadratic** and render out to see the difference.
6. Do you see any shadow? To project the shadows from this spot light, look for **Shadow** section and turn on **Use Depth Map Shadows** in **Depth Map Shadow Attributes**. This option also prevents the light pass through by the first object it encounters.
7. To reveal the appearance of the light beams (i.e., fog or dust), click **Map** button next to **Light Fog** in the **Light Effects** section. Render out to see the effect. Adjust the color in **Light Fog Attributes** to generate an interesting color mood.
8. Other attributes to consider;
 - **Cone Angle**
 - **Dropoff** (intensity control)
 - **Penumbra** (softness along the edge)

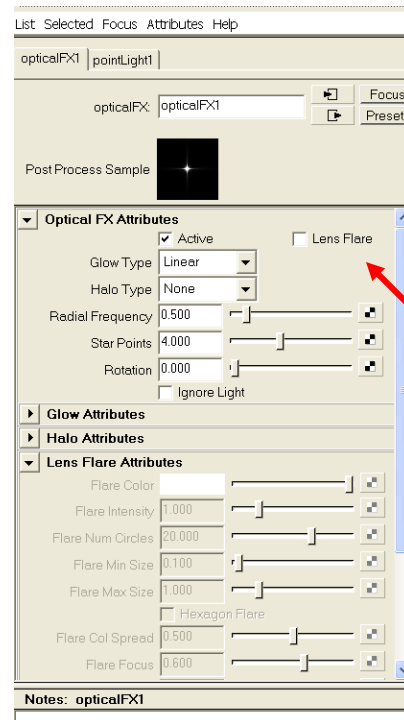
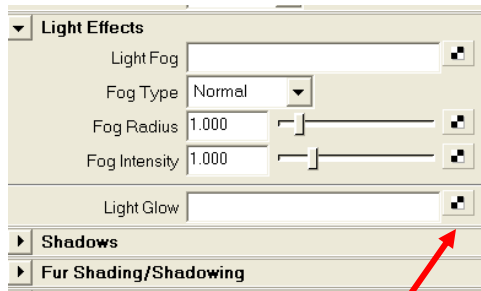




Decay Regions & Map

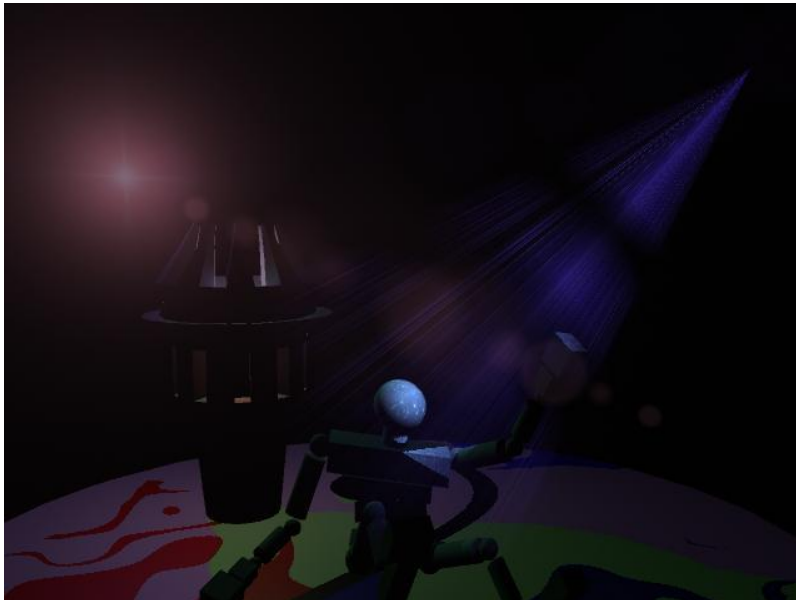
9. To have more precise control over decay, Maya provides Decay Region tool that could divide the region into sub-regions. Let's first turn on the **Use Decay Regions in Decay Regions**. Also, display the Decay Region manipulators (**Display > Rendering > Camera/Light Manipulator > Decay Regions**).
10. Using **Move** tool, you could interactively reposition the distance of each region. Click and drag each region (manipulator ring). Play with **Intensity** value to generate different moods.
11. Rather than using simple color for the light, you could map color into an image file. Click **Map** button next to **Color** attribute, select the File button, click **Folder** icon next to the **Image Name** then load an image file. (i.e., nebula). Test the effect.
12. If you want to remove the map, RMB click on the Color label and select the **Break Connection**.





Point Light & Lens Flare

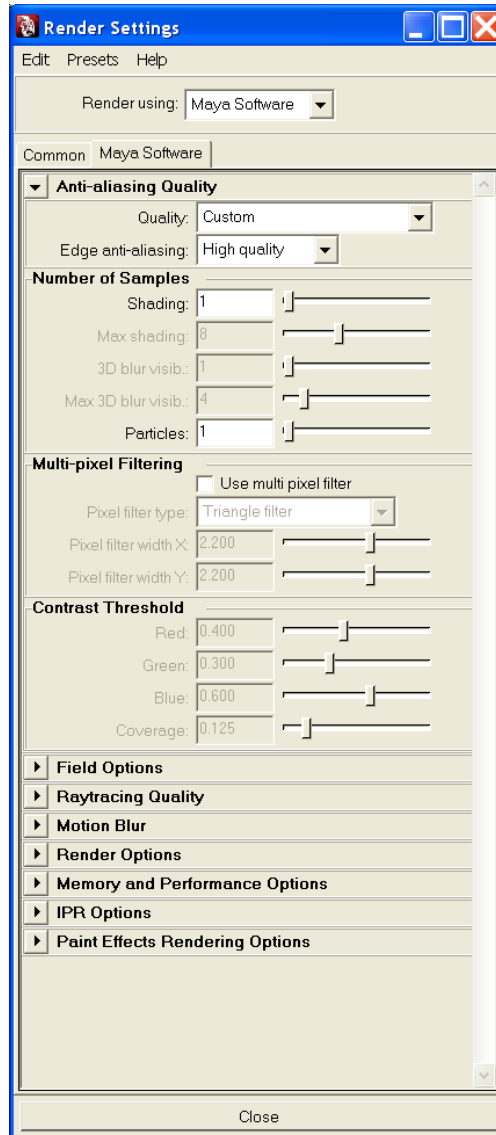
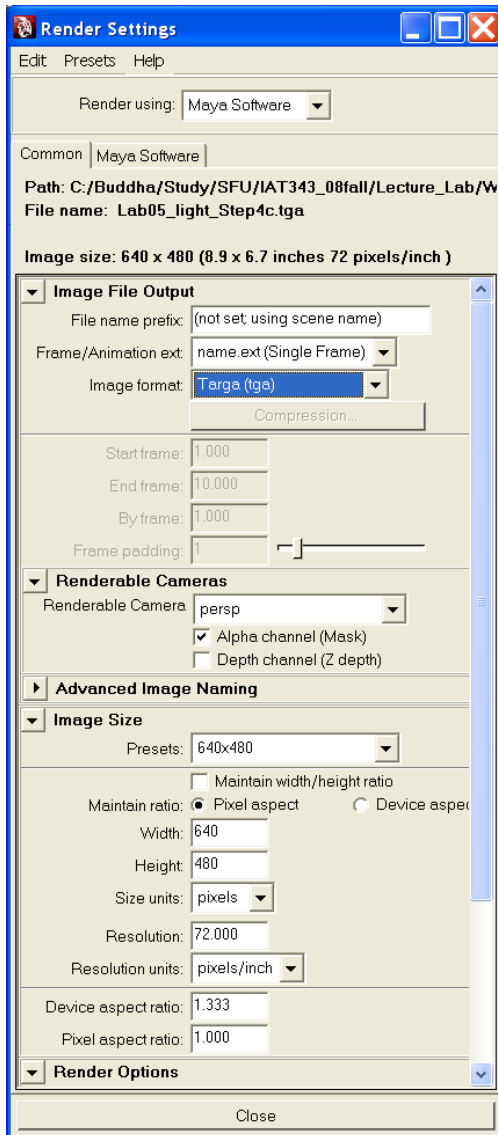
13. **Light Glow** attribute provides optical effects (Glow, Halos, and Lens Flares). Rather than adding this effect on the existing lights, let's create another light (**Create > Lights > Point Light**). Position this light above the torch.
14. In the **Light Effects** attributes, click **Map** button next to **Light Glow**. It opens Optical FX attributes that include **Glow**, **Halo**, and **Lens Flare**. Turn on **Lens Flare** option. Render it out to see the effect. You could lower the **Intensity** of the **Point Light** Attributes.
15. Adjust number of attributes to make the scene more appealing.
 - Glow Type
 - Halo Type
 - Halo Color
 - Halo Intensity
 - Etc...



Area light

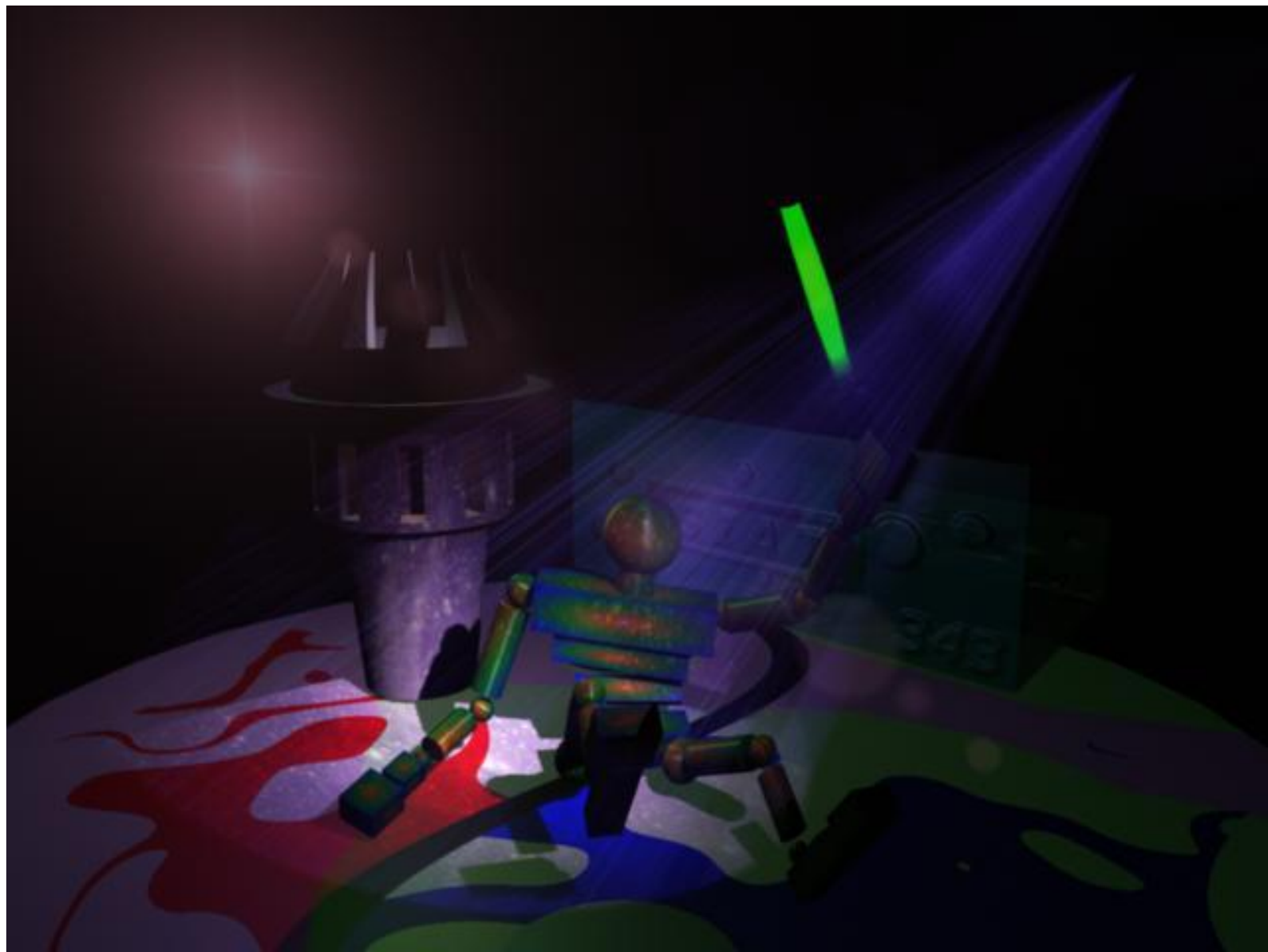


16. In *Star Wars*, Luke Skywalker uses the lightsaber. To simulate this effect, first create a stick object (i.e., sword or any weapon of your choice).
17. Add an Area light (**Create > Lights > Area Light**). Reposition the area light close to the center of the light. Transform (i.e., scale) the shape of the area light similar to the stick.
18. Adjust number of attributes (i.e., color) to make the stick interesting.



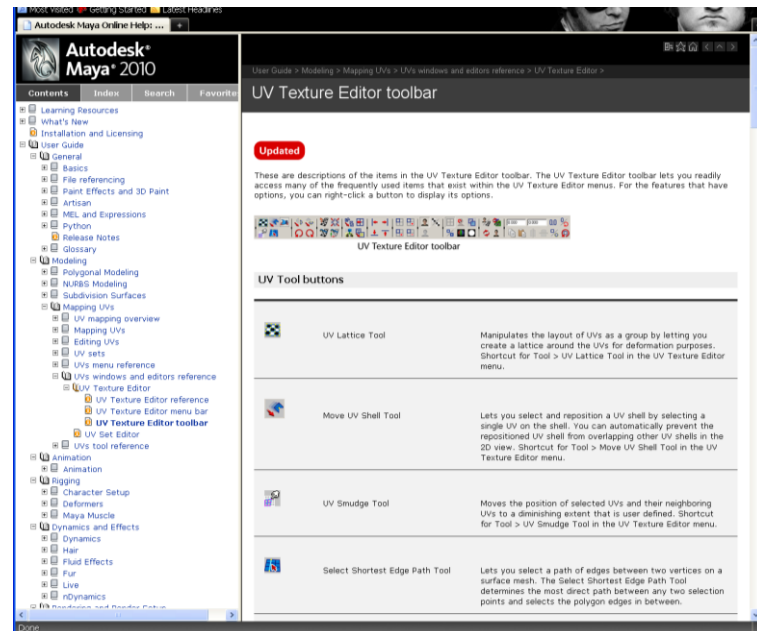
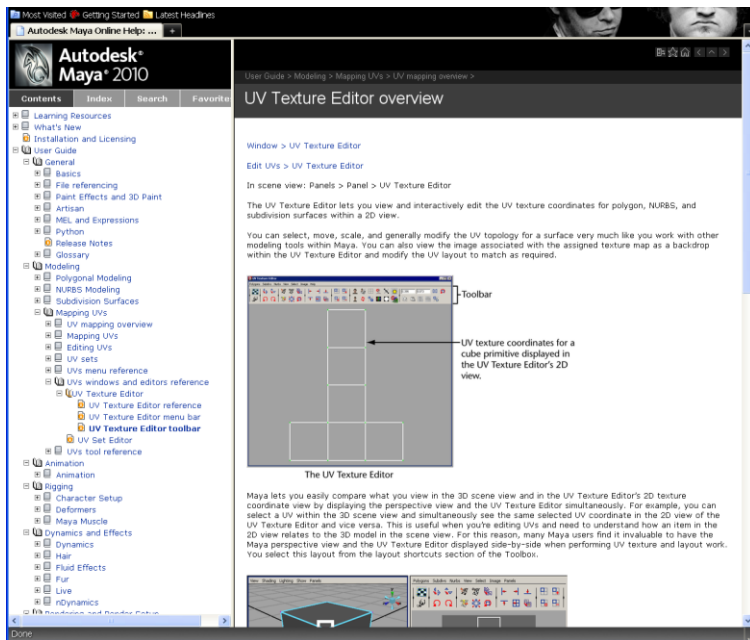
Rendering Quality

16. The **Render Settings** menu has a set of global attributes to control the quality of rendering. Also, it provides different settings relate to rendered images. To open this menu, choose **Window > Rendering Editors > Render Setting**.
17. Under **Common** tab, choose a type of **Image formats**, **Image Size** and **Renderable Camera**.
18. Make it sure the location of a saved image (**Edit > Change Project Image Directory**).
19. Switch to **Maya Software** tab. To improve the quality of rendered images, Maya provides **Anti-aliasing Quality** option. Change **Quality** option and **Edge anti-aliasing** option to higher quality.
20. Render under different settings and compare the image qualities of the rendered images.

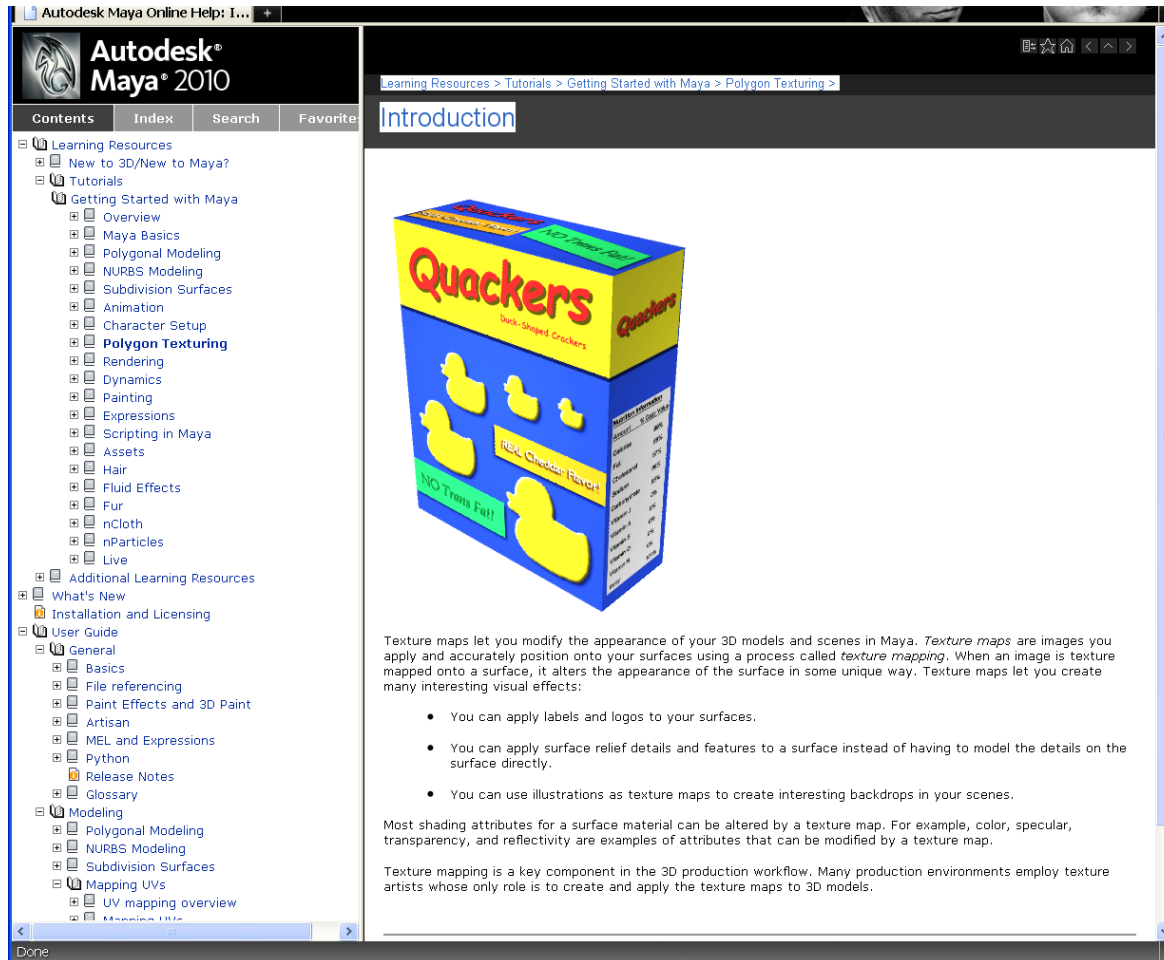


User Guide > Modeling > Mapping UVs > UV mapping overview > UV Texture Editor overview

User Guide > Modeling > Mapping UVs > UVs windows and editors reference > UV Texture Editor > UV Texture Editor toolbar



Learning Resources > Tutorials > Getting Started with Maya > Polygon Texturing > Introduction



The screenshot shows the Autodesk Maya 2010 Online Help interface. The left sidebar contains a navigation tree with the following structure:

- Autodesk® Maya® 2010
 - Contents
 - Index
 - Search
 - Favorite
 - Learning Resources
 - New to 3D/New to Maya?
 - Tutorials
 - Getting Started with Maya
 - Overview
 - Maya Basics
 - Polygonal Modeling
 - NURBS Modeling
 - Subdivision Surfaces
 - Animation
 - Character Setup
 - Polygon Texturing**
 - Rendering
 - Dynamics
 - Painting
 - Expressions
 - Scripting in Maya
 - Assets
 - Hair
 - Fluid Effects
 - Fur
 - nCloth
 - nParticles
 - Live
 - Additional Learning Resources
 - What's New
 - Installation and Licensing
 - User Guide
 - General
 - Basics
 - File referencing
 - Paint Effects and 3D Paint
 - Artisan
 - MEL and Expressions
 - Python
 - Release Notes
 - Glossary
 - Modeling
 - Polygonal Modeling
 - NURBS Modeling
 - Subdivision Surfaces
 - Mapping UVs
 - UV mapping overview
 - Mapping UVs

The main content area displays the title "Introduction" and a 3D rendering of a Quackers cereal box. The box is blue and yellow with the "Quackers" logo in red. It features three yellow duck characters and the text "Suck-Shaped Crackers". A label on the box reads "REAL Chicken Flavor". A green banner at the bottom of the box says "NO TRANS FAT!".

Texture maps let you modify the appearance of your 3D models and scenes in Maya. Texture maps are images you apply and accurately position onto your surfaces using a process called *texture mapping*. When an image is texture mapped onto a surface, it alters the appearance of the surface in some unique way. Texture maps let you create many interesting visual effects:

- You can apply labels and logos to your surfaces.
- You can apply surface relief details and features to a surface instead of having to model the details on the surface directly.
- You can use illustrations as texture maps to create interesting backdrops in your scenes.

Most shading attributes for a surface material can be altered by a texture map. For example, color, specular, transparency, and reflectivity are examples of attributes that can be modified by a texture map.

Texture mapping is a key component in the 3D production workflow. Many production environments employ texture artists whose only role is to create and apply the texture maps to 3D models.