For all of these tutorials you will need Blender. You may download it here if you need.

Blender 2.28

The Reblended.com Learning Source

The Basics

- Blender's Interface
- The Blender Windows
- Navigating in 3D space
- Saving
- Building a Castle
- Texturing a Castle

Warning: You might need to complete the above tutorials before you continue down this page!

Advanced Tutorials

- Appending files in Blender
- Building a spiral stairs
- Candle modeling
- Controlling dynamic objects
- Environmental maps
- Fireworks
- Controlling flocks of objects
- Game engine

 Game textures

 Global Illumination

 Head creator plugin

 Ikas (Bones)

 Jump!

 Particles

 Radiosity

 Refraction

 Shaded Particles

 Spotlight

 Toonshading

 UV mapping

Key:

 - Novice

 - Intermediate

 - Experts

---

If there are any tutorials that you would like to see here email webmaster@reblended.com.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=3

Tutorials
2000 07 21

User Interface

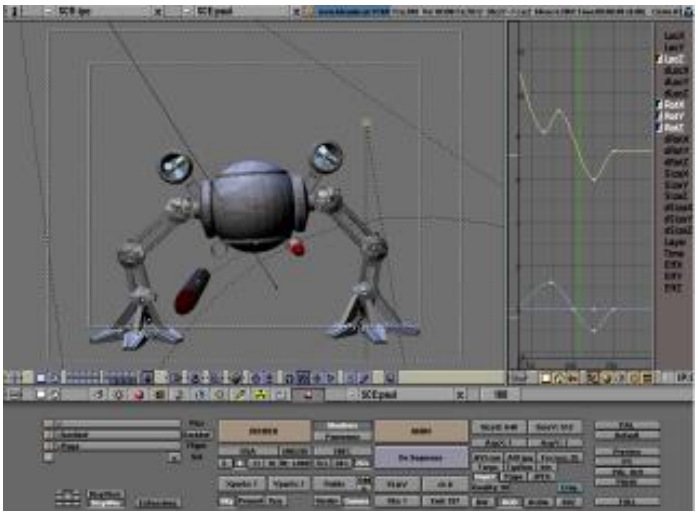
Bart Veldhuizen

id3

Introduction

If you are reading this tutorial, you are probably feeling like I did exactly one year ago. I had found this extremely interesting looking 3D application (which other users were raving about), but the user interface completely baffled me. I couldn't find the quit function (I had to kill the application instead of exiting it), I saw buttons that seemed to react differently each time I clicked on them and every time something interesting happened, I could not reproduce it.

In the weeks after that, I slowly found out the basic principles behind the Blender user interface. And though it is non-standard, it became clear that it was a very consistent system - it would let me use the same functions in a number of completely different situations.



Blender in Action.

This tutorial will save you weeks of frustration by explaining the basics of Blender's user interface. It will not explain every button or even every window in detail (that is where the Blender manual comes in), but instead let you see the basic idea behind it.

After finishing this tutorial you are ready to work with the rest of the book. Like me, you will find out that Blender's user interface really grows on you; it is one of the most efficient and well-thought out interfaces I have worked with.

Here is what I found to be the Golden Rule of Blender:

"Keep one hand on your keyboard and one on your mouse."

Even while Blender's user interface may be a bit daunting at first, I hope that working with this book will teach you that it is actually a very intuitive one. It was written entirely for working efficiently and quickly and this means that a lot of functions are accessible both by using the keyboard and by using the mouse. Working with hotkeys instead of menus is harder to learn, but the rewards are great.

After a while working with Blender becomes a second nature. Often, my girlfriend watches in amazement at the speed with which I can handle objects, change views, navigate your way around your scene in Blender. Blender has really turned my keyboard and mouse into a glove which reaches into 3D space.

But it all starts with the Golden Rule - remember him while you are working your way through this book!

What am I looking at?

When you first start Blender, your screen will look like this. The screen is divided into three parts; on the top you see the Info Window. Among other things, you will see Blender's version number and some statistics about your current scene.

An empty Blender scene.

Next is the 3D Window. This window type is used for all the editing in your 3D world. The pink square in the middle is the standard plane. It is pink because it is currently selected. The black triangle on the bottom is the camera. The gray lines are the grid lines of the 3D world - you can use them to align your objects. The cross hairs with the white and red colored circle is the 3D cursor. You can place it anywhere in 3D space. It is used to determine where new objects are placed, but it can also act as a center for rotation or scaling.

The window on the bottom is called the Buttons Window. In this window, you can edit a variety of information about your scene: materials, lights, animation settings, rendering settings - it is all in here.

The window header.

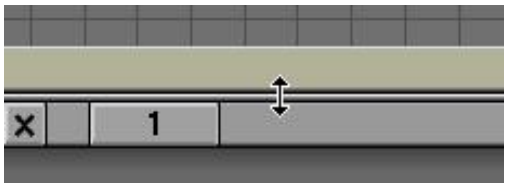
Each window has a Header. For the 3D window the header is located at the bottom. You can switch the position of the header by right clicking on it and confirming the requester that pops up ('Switch header'). Pay special attention to the button on the far left - each window has this icon. This is also the one that completely eluded me in the beginning.

In Blender terms, it is called an Icon Slider button. If you click your left mousebutton on it and drag left or right it will change, revealing a number of different icons. Each icon corresponds to a different window type. I will explain the different types later.

You can also click on the left and right half of the Icon Slider to change the window type.

The contents of the Buttons Window can be changed by clicking on one of these buttons. In fact, it is a lot like the Icon Slider button that I discussed, but now all the options are visible. The meaning of the different buttons is explained in 'the Buttons Window' below.

Selecting a Button Window.



Moving a window border.

Configuring the screen your way

Blender's screen can be reconfigured in any way you like. To start with, you can change the size of the different windows by placing the mouse pointer over one of the lines separating the different windows. The cursor changes shape to indicate you can now change the window size. Left click and drag to change the window size.

Adding a new window is similar: move the mouse pointer over a window border and press the middle mouse button. A small requester will pop up now. You can confirm splitting the screen by left clicking on the text 'Split', or by pressing Enter. The window which has the focus and has a yellow header (in this case, the top window) is the one that will be split - approaching the window border from below will split the bottom window.

Splitting a screen.

A line indicating the position of the split is now shown. You can move it to the desired position and left-click to confirm the location.

Cancel the action by moving your mouse pointer away from the requester, or by pressing Esc.

A Blender screen with four 3D windows.

Of course, splitting windows works on vertical window borders, too. In this example, I have created three additional 3D Windows by splitting the default 3D Window three times. Each window is now independent from the others. This way, you can create the traditional 4-view window (top, front, right and camera) in Blender. More about this below in 'the 3D Window'.

Press Ctrl ↑ to make the current window full screen. Press it again to restore the previous view.

Removing windows is very much like splitting them; with the mouse pointer over a window border, right-click. Confirm the requester. Like splitting, the direction from which you approached the border (or rather, which window has the focus before joining) determines which window 'survives' the joining.

Joining two windows.

ADD	Load File As	F1
VIEW	Load Last	c|o
EDIT	Append File	shift+F1
OBJECT		
OBJECT	Save File as	F2
MESH	Save	c|w
CURVE	Save Image	F3
KEY	Save VRML	c|F2
RENDER	Save DXF	shift+F2
FILE	Save VideoScape	a|w
	Quit Blender	q

The toolbox.

The Toolbox - Adding a Sphere

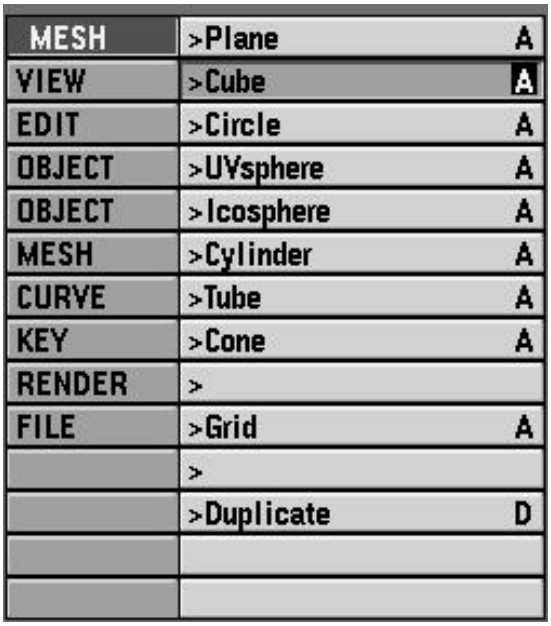
Almost every Blender function can be accessed by using the keyboard or the mouse. The Toolbox contains most of Blender's functions. You can call up the toolbox either by clicking on the toolbox icon in the top-right of the Blender screen, or by pressing [ ].

This is how the toolbox works: The dark gray buttons on the left are function categories; moving your mouse pointer over them will change the button list on the right. Move your mouse pointer to the right and click on a function button to activate that function. Each button also shows the corresponding keyboard shortcut. In the shortcut notation, c|<button> means holding down [Ctrl] and pressing the listed button. Similarly, a|<button> uses the [Alt] key as a modifier.

Clicking some buttons will reveal a third layer of functions. For example, if you wish to add a sphere to your scene, first select 'Add' from the categories list in the Toolbox. Next, left-click on the button labeled 'Mesh'.

ADD	Mesh	>>
VIEW	Curve	>>
EDIT	Surface	>>
OBJECT	Text	
OBJECT	MetaBall	
MESH	Empty	A
CURVE		
KEY	Camera	A
RENDER	Lamp	A
FILE	Ika	A
	Lattice	A

Starting to add a mesh.



Adding a cube.

The button list on the right now changes to offer you a choice between the different primitives that Blender knows. left-click on 'UVsphere' to add a sphere to your scene.

When adding objects like a sphere, Blender wants to know which resolution you require. A sphere with a high resolution looks smoother than a low-resolution sphere, but it also requires more memory and is slower to manipulate and render.

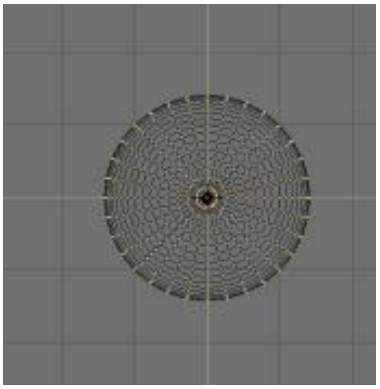
Parameter for a sphere.

These buttons work like the Icon slider you saw earlier: left-click on the 'Segments' button and drag left or right to increase or decrease the value. Also, you can click on either side of the button to change the value step by step.

Editing a parameter.

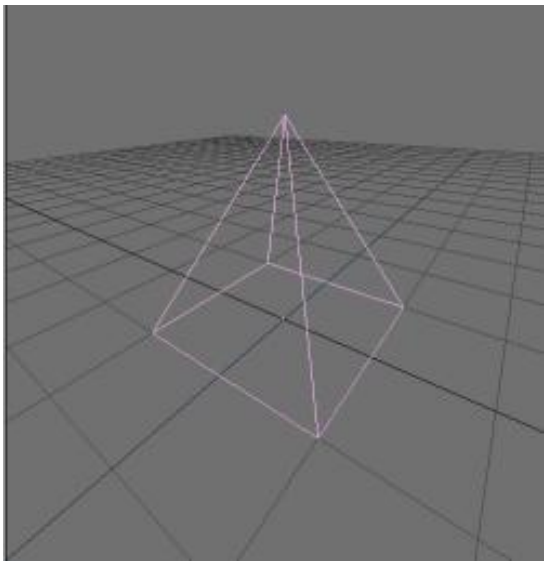
Yet another way to change the value is to hold down Shift and left-click the 'Segments' button. This will change the button into a text input. You can now type a value into the field directly. Press Enter when you have entered the correct value.

Finally, click on the OK button or press Enter to set the value of the 'Segments' variable.



A mesh sphere in edit mode.

In the case of the sphere, you will need to set a second parameter, 'Rings'. After doing this, a sphere will show up at the location of your 3D Cursor. Instead of showing up in pink or black, the sphere is drawn with small yellow dots. This mode is described in the paragraph 'Edit Mode'.

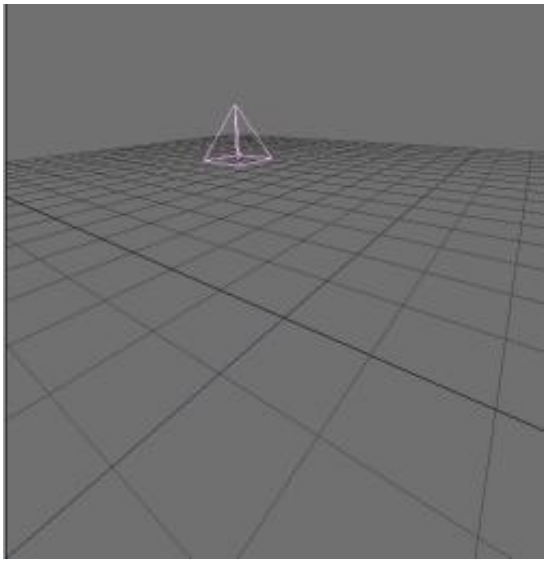


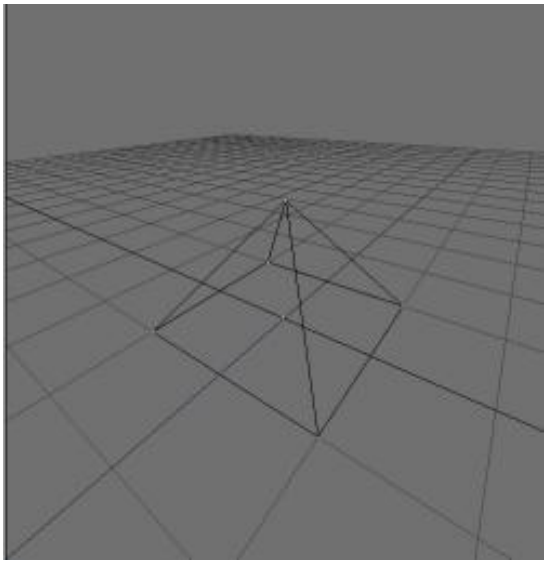
About Edit mode

When you are working in 3D space, you can basically perform two types of operations: operations that affect an object as a whole and operations that affect the geometry of an object, but do not affect the object's global properties An example of the former is moving an object to another location. An example of the latter is shaping a nose on a face.

In Blender, you have to indicate which of the two you want to use. This is where edit mode comes in.

This is an example of an object that is not in edit mode. Operations like translating (G), rotating (R) or scaling (S) affect the object as a whole. For example, translating moves the entire object somewhere else.





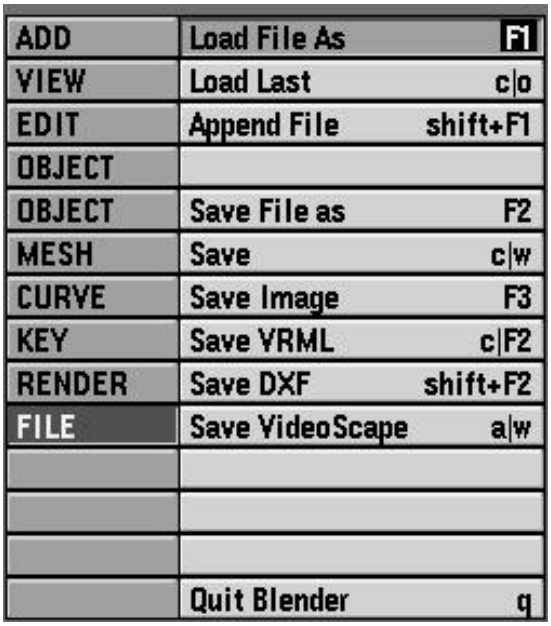
In contrast, when you select Tab you enter edit mode. The object is redrawn and the separate points of which is consists are drawn. (These points are known as vertices, by the way). If you select a vertex by right-clicking on it, you can then translate only this point, while leaving the others unaffected. In this way, you change an object's shape.

If you select more than one vertex, you can also rotate or scale them. In this case, the vertices are rotated or scaled around their median point.

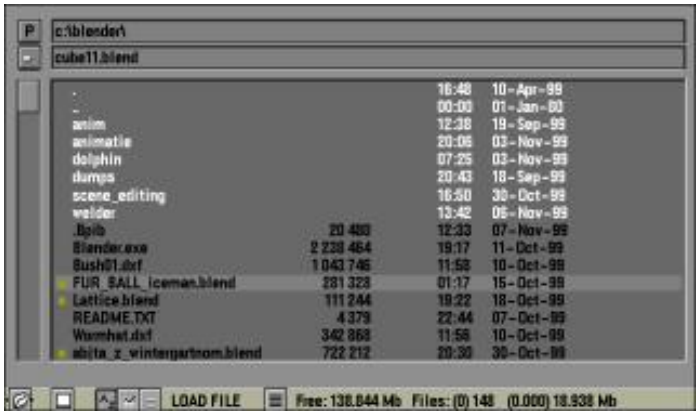
Loading and saving your work

By now you have probably created a Blender scene which you might want to save to disk, or you would like to take a look at the .blend files that we have included on the CD-Rom that comes with this book. Blender has a lot of file loading and saving options - even to the point that you can use Blender as a file manager.

In this part I will briefly describe the basic file loading and saving mechanisms. As always, these functions can be accessed either from the Toolbox or by using a hotkey.



Accessing the file functions in the toolbox.



The Load File window.

Loading a Blender file

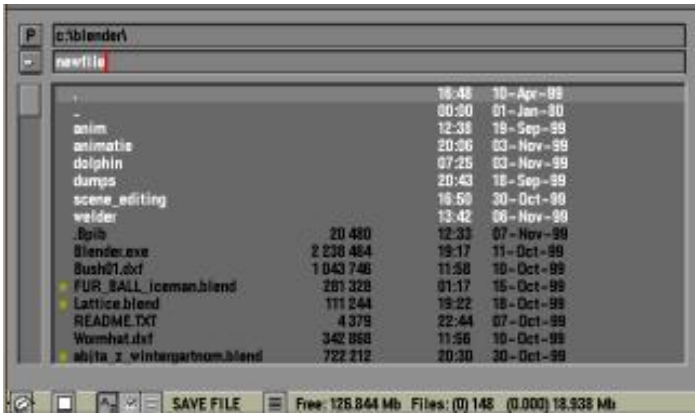
Call up the File Window by pressing F1. In this window, files are shown in black and directories in white text. The .blend files are also indicated with a small yellow square in front of them.

The first method to load a file is by left-clicking on a filename. This will copy the name to the filename box at the top of the window (the second textbox from the top). Pressing Enter will then actually load the file. Alternatively, you can also middle-click on a file which will cause it to be loaded immediately.

Navigating to the parent directory is done by clicking on the '..' directory in the list, by clicking on the 'P' button at the top of the window or by pressing P. Sometimes you may want to refresh the file list; for example when another program has been writing or changing files in the current directory. In that case, click in the '.' directory or press the dot key.

Saving a Blender file

Open the File Window by pressing F2. This window works in the same way as above. Additionally, you can now enter a new filename in the filename box. To do this, left click in the filename box and enter a new name.



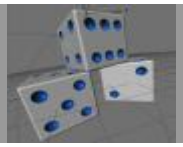
The Save File window.

Use the keys numpad + and numpad - to automatically increase or decrease the versionnumber of a file before loading or saving

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=1

Tutorials
2000 07 20

The Blender Windows



Bart Veldhuizen

id1

A short overview of functionality

This tutorial gives a summary of the most important windows and ButtonsWindows in Blender. It does not try to be complete nor to explain each window in detail. Much of the functionality of these windows is described in tutorials, both on this site and on web pages from Blender users.

The DisplayButtons (F10)

Once you are ready to render your image or save it to disk, you need the DisplayButtons window. In this window you can control all the parameters that have to do with items as rendering quality, rendering size, animation length and animation filenames.

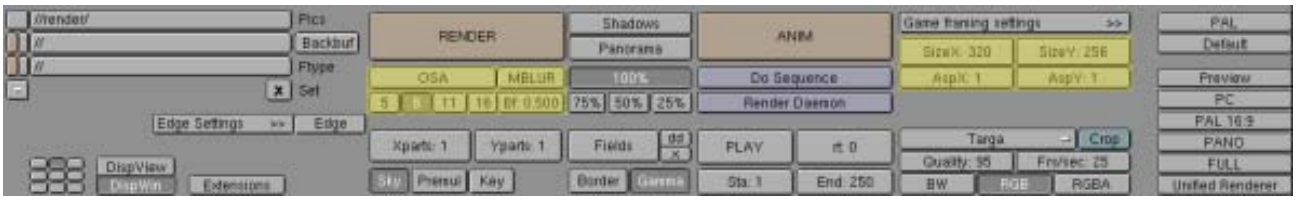


Image size and quality

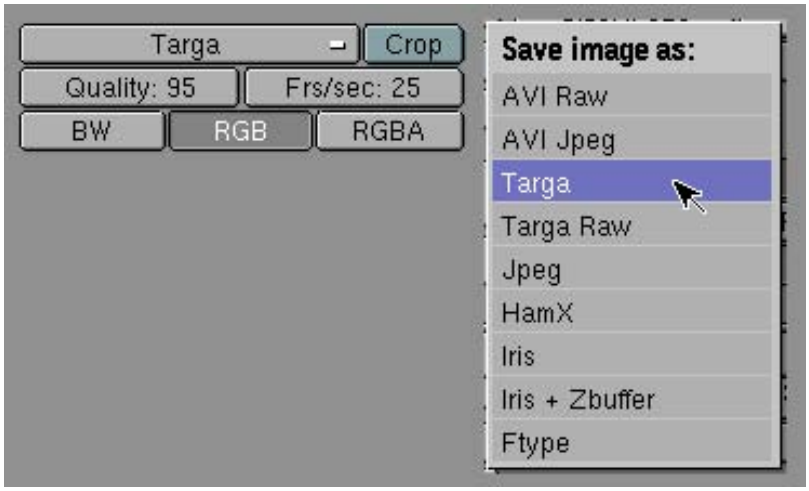
To define the image size you can either select a preset value from the row of buttons on the far right or set an image size yourself by changing the SizeX and SizeY values. For preview rendering, you can also render the image at 25%, 50% or 75% of the final output size.

Turn anti-aliasing on or off with the OSA button. Control the image quality with the oversampling buttons 5, 8, 11 or 16. A higher value will result in a better image quality at the penalty of a longer rendering time.

Saving animations.

Start rendering by selecting the 'Render' button or by pressing the F12 key. A separate window will pop up. Esc aborts rendering. F11 will toggle the rendering result window.

Press 'Anim' to render an entire animation. When rendering an animation, enter the basename of the animation frames in the 'Pics' field. Each file will be automatically named <basename>+framenumber. For example: frame0001, frame0002 etc. After rendering, press the 'Play' button to play back your animation.



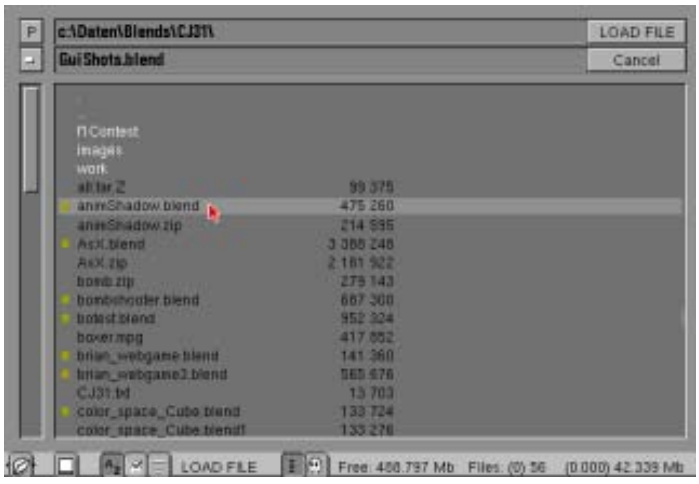
Choosing image and animation formats

Select a file format with the MenuButton labeled 'Targa' here. A pop-up menu lets you choose the desired format. For animations, you can also choose 'AVI raw' and 'AVI jpg' and define the number of frames per second for these files.

The Windows Mediaplayer has problems playing back Blender's AVI-raw and jpg files, but the Quicktime4 player plays them back correctly.

Animation length.

To determine the start- and endframe of your animation, change the Sta: and End: values. After you have rendered your animation press Play to play it back.



The FileWindow

The FileWindow

Hotkeys: F1 for loading, F2 for saving.

Left click to select a file and press Enter to load it, or middle click to both select and load it. Alternatively you can use the "LOAD" (or "SAVE") buttons in the upper right corner of a FileWindow.

The FileWindow in Blender is general and will be called for all file operations.

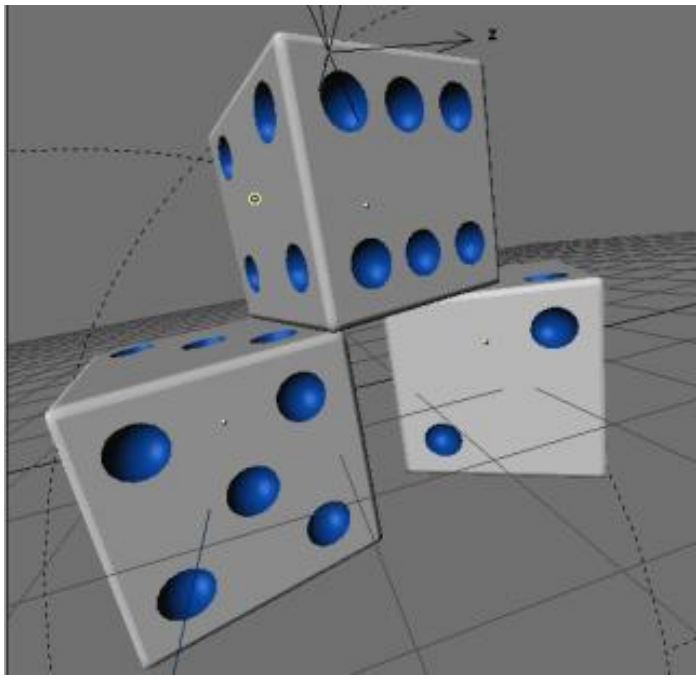
To save a file, enter a name in the filename field (the second one from the top) and press Enter to confirm the filename. Press [enter] again to save the file. If you have already entered a filename earlier, pressing F2, Enter will save, too.

Entering a new directory name in the directory box (the one on the top) and pressing Enter will create the new directory for you after asking confirmation.

The ImageSelectWindow

This window type works exactly like the file load window, but all recognized image files are shown as thumbnails. An example of where this window is used is in the Texture Buttons window, for loading image texture maps.

The Image Select Window



The 3D Window

The 3D Window

This window allows you to manipulate the objects in your scene directly by dragging (G), rotating (G) or scaling them (S).

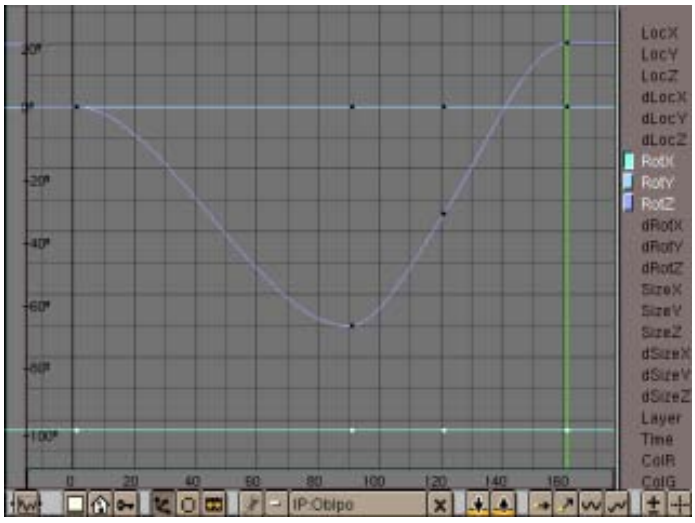
Change between top, front and right view with numpad 7, numpad 1 and numpad 3. Toggle perspective mode with numpad 5, and change to camera view with numpad 0. Toggle between wire, solid or shaded drawing with Z and Ctrl Z

You can navigate through 3D space by dragging with the middle button; this will rotate your view. Dragging with middle button and holding the Shift key will translate the view, and holding down Ctrl will allow you to zoom.

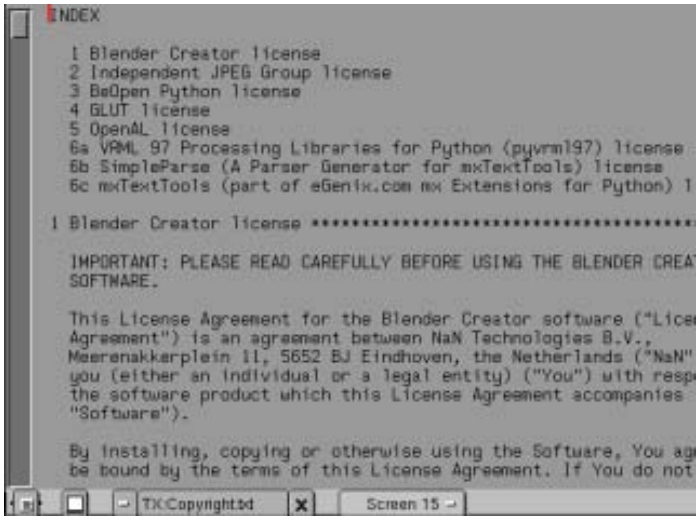
The IPOWindow (SHIFT-F6)

In this window you can edit the animation curves that are associated to different object properties, like their position and rotation, but also their color or layer. The different object properties or 'channels' that are available are shown on the right. You can select channels by left clicking on them - Shift will extend the selection.

Just like in the 3Dwindow, enter edit mode by pressing Tab. You can now edit the individual vertices of the curves or change their curvature.



The IPOWindow



TextWindow

The TextWindow (SHIFT-F11)

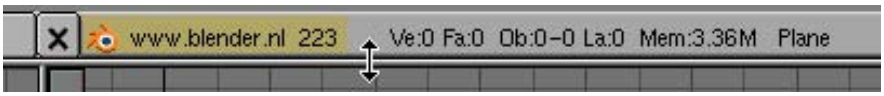
The TextWindow is mainly used for writing Python scripts, but you can also use it for other things.. It is often a good idea to keep notes of your animation projects. Information that I always forget (especially after a couple of weeks!) are the maps that I have used, the scenes that I have defined, important frame numbers, well, you get the picture.

A big advantage of using the TextWindow is that the notes are stored inside the .blend files - this way you can never loose your notes again!

When creating a new TextWindow, you have the choice of creating a new text object, or to import an existing text file. In that case, a FileWindow will be opened and you can select a file as usual. Of course, you can have more than one text object in your Blender file.

The Info Window

The Info Window does not have a hotkey. Instead, it is 'hidden' at the top of your screen. To reveal it, drag down the top window border on your screen as indicated in the picture.



Drag the window edge to reveal the InfoWindow

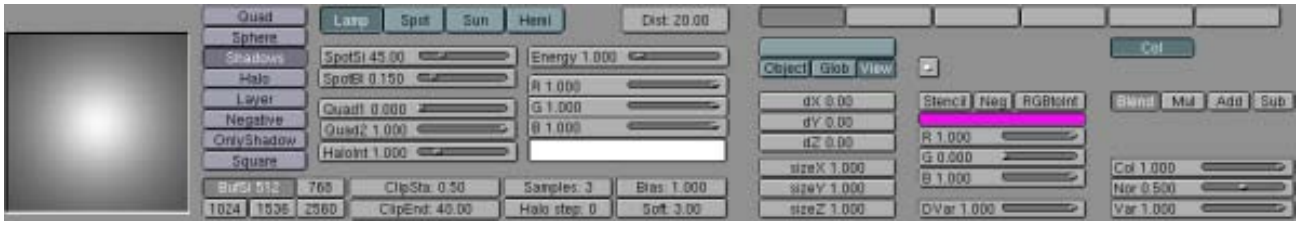
InfoWindow settings.

The Info Window shows you the settings of Blender. Among other things, you can tell Blender where to store autosave files, when to save them (use the 'Time' value) and how many versions to keep on disk. Here, you can also activate Blender's tooltips function. Tooltips will appear in the right corner of the header of the Info Window, right next to the Blender URL and version number.

You can make your settings permanent by pressing Ctrl U and confirming the requester that pops up. (Please note, that all the tutorials on this site assume that you have not changed the default Blender settings file!)

Important note for owners of a two-button mouse: activating the 2-mouse option will cause Blender to simulate a middle click when you hold down Alt and left click.

Lamp Buttons (F4)



Lamp Buttons

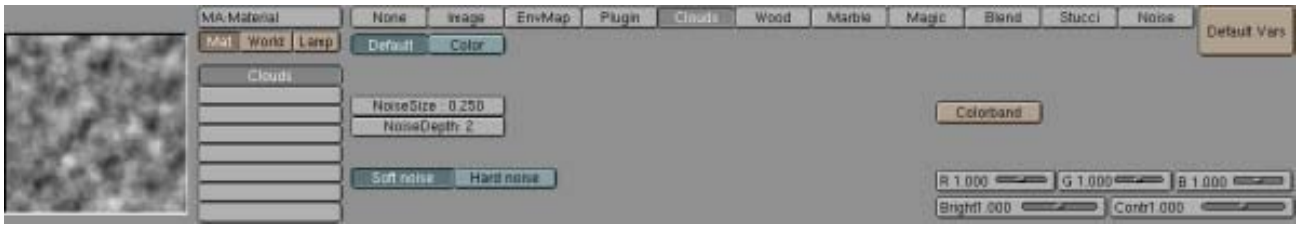
The Lamp Buttons will only show when a lamp has been selected. With these buttons, you can change all the parameters of lamps, like their color, energy, type (regular lamp, spotlight or sun). You can also control the quality of shadows by manipulating the clipping values and shadow buffer sizes.

MaterialButtons (F5)

MaterialButtons are only shown when an object with an Material has been selected. To create a new Material or browse in the scene-materials use the MenuButton in the MaterialButtons header. This window allows control over properties as object color, shininess, transparency, textures, texture types, texture projection methods.

More information about materials and texture maps is available in Texturing a Castle.

TextureButtons (F6)



Texture Buttons

In this window you can select several types of textures for use in a material, light or world setting. The available types are:

Image: You can load an image as a 'texture map'. This image will then be projected on an object in the way that you define (for example using planar or spherical projection). Projection method is a material property.

Procedural: Clouds, Wood, Marble, Magic, Blend, Stucci and Noise. These textures are predefined and have a number of parameters that you can adjust. Procedural textures are also truly three dimensional. An example of this would be a block of wood: if you cut out a part of it, the wood texture needs to continue on the inside in a realistic manner.

Plugin: You can also program your own piece of code to use as a procedural texture. It is similar to writing a sequence editor plugin. More information about this can be found in the help section of this site.

Environment map: Environment maps are used to simulate reflections of the environment in an object. To achieve this, Blender calculates six images from the object's viewpoint. These images are then combined to compute a reflection.

Animation Buttons

AnimationButtons

In the left part of the Animation Buttons window you can set properties for things like curve following, automatic object duplication and object tracking.

The middle part of the window contains the interface to object plugins like particle systems and the wave effect. Pressing the 'New Effect' button and selecting an effect from the effects list on the right will attach a new effect to the currently selected object and display the list of options for this effect.

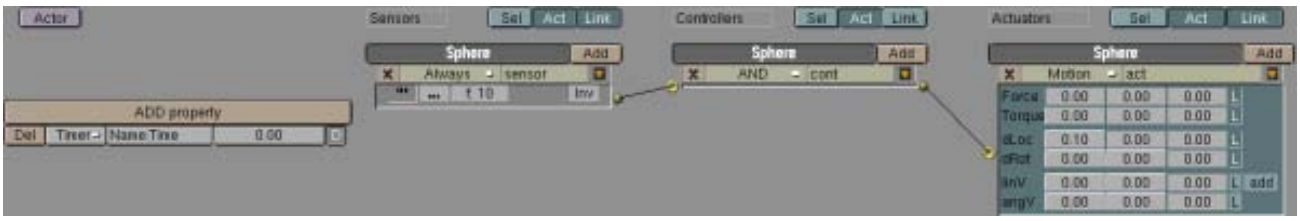
The standard effects are: particle system, wave and build.

WorldButtons

WorldButtons

In the WorldButtons you can define settings that are scene-global. These are for example the mist (fog) settings, rendering of stars, the color and textures of the horizon and zenith.

RealtimeButtons (F8)



The RealtimeButtons are Blenders editor for defining interactive realtime 3D graphics. Here you can edit settings for the physics of realtime objects, and connect objects with game-logic. Check our numerous tutorials on creation of real-time content!

EditButtons (F9)

EditButtons

The buttons that are shown in the EditButtons window depend on the kind of object that you have currently selected. Each type of Objects has its own specific set of Buttons and settings. Meshes for example have Buttons for manipulation of vertices and Curves have Buttons to edit the resolution and the order of the Curve.

Display Buttons (F10)

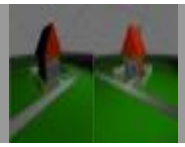
DisplayButtons

Using the Display Buttons, you can control the way Blender will render your images. Set the image size with the SizeX and SizeY buttons and control the image quality by setting an OSA (Over SAmpling) value. Higher values result in better image quality, but a longer rendering time. Select the 'Render' button to render a still image, or the 'Anim' button to render a range of images. (For animations, the 'Sta' and 'End' values determine the length of the animation).

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=6

Tutorials
2000 07 20
id6

Navigating in 3D Space



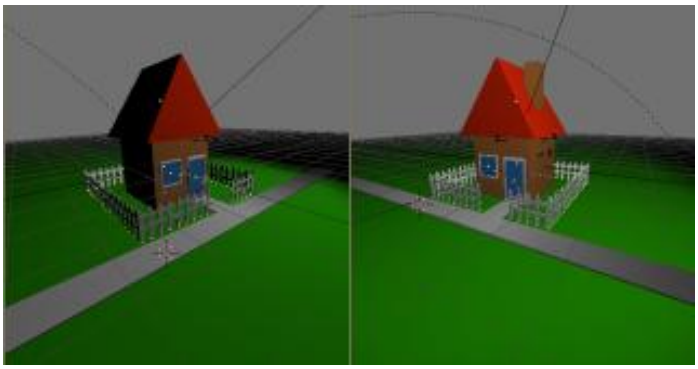
Bart Veldhuizen

The 3D window is where you will spend most of your time in Blender. In this window, you can rearrange the objects in your scene, edit their individual vertices (vertices are the points that make up an object), define animation, add lights and cameras, ...

If you own a mouse with two buttons instead of three , skip ahead and read about the Info Window in 'The Blender Windows: a summary'

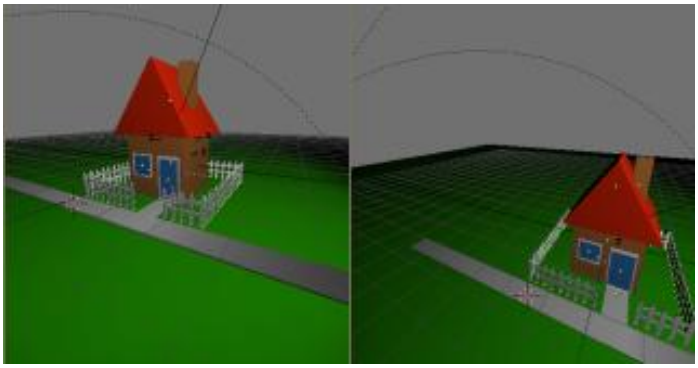
The first thing to know about the 3D view is how you can navigate through it. I am not talking about moving objects, but about moving your viewpoint - how can you examine objects? How does an object look from another side? Do the different parts have the correct color? Did I add a little detail correctly?

Blender uses a very simple yet intuitive method for changing the view in 3D. To get started, press and hold the middle mouse button and drag your mouse around. Do you see what happens? Without any modifiers, middle mouse rotates your 3D view.



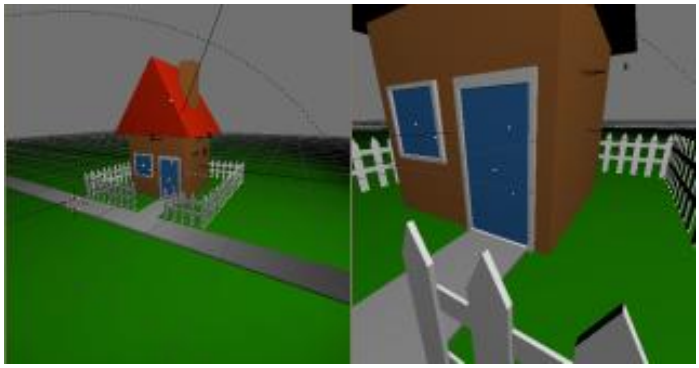
Rotating the 3D view.

Now try the same but hold down the Shift key before you click the mouse button. The view is now translated instead of rotated.



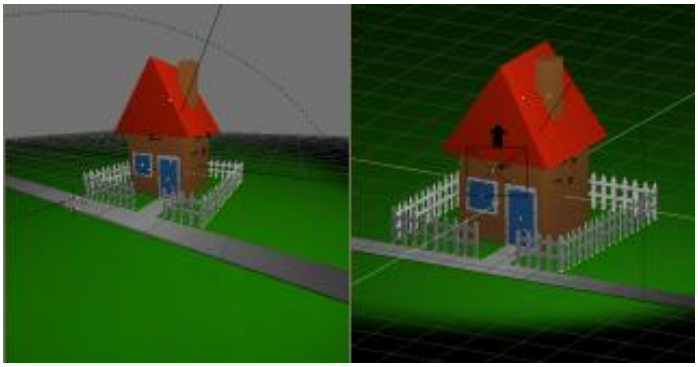
Translating the 3D view.

Finally, the Ctrl key allows you to zoom in and out.



Zooming in the 3D view.

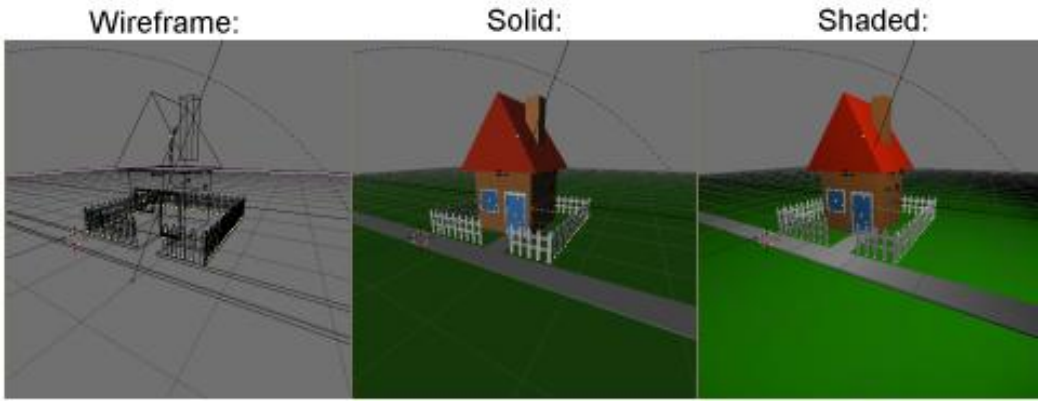
To toggle between perspective and orthogonal view, press numpad 5.



Perspective/orthogonal view.

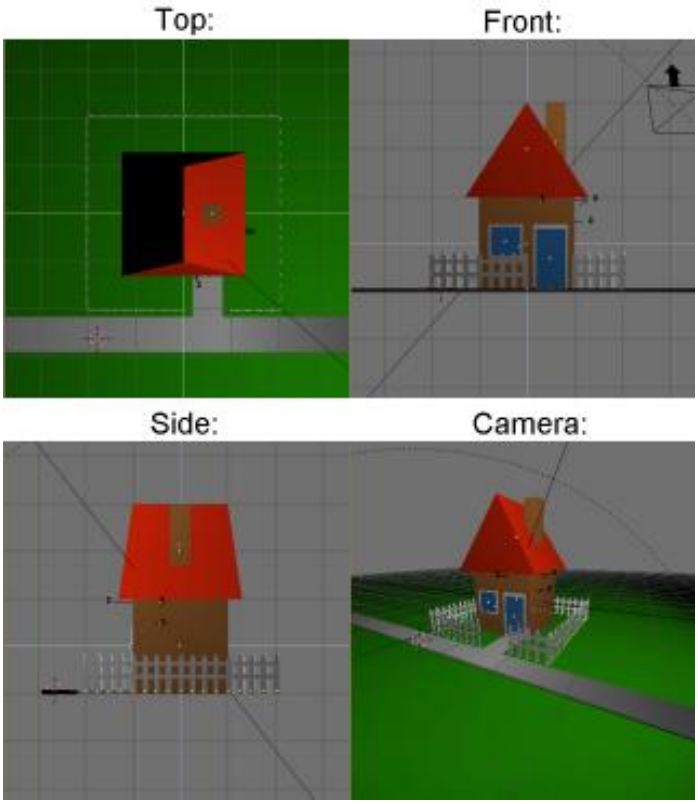
In the 3D window, you can display your scene in a number of modes. Especially if you don't have 3D hardware this can make a huge performance difference. To toggle between wireframe and solid view, press Z. Solid view can display a scene with some shading even though you have not yet defined a lamp.

Switch to (real) shaded view with Shift Z. If you have not defined a lamp yet, the scene will be completely black.



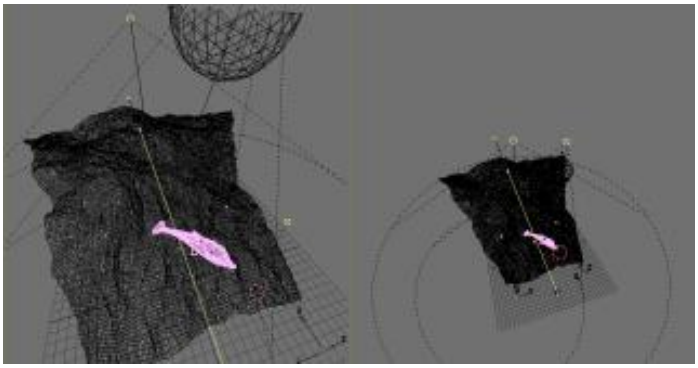
Wireframe/solid/shaded.

Aside from navigating around 3D space to look at your objects, you can also use a few hotkeys to quickly switch to standard views. Use numpad 7, 1, 3 and 0 to switch to top view, front view, side view or camera view.



Different views.

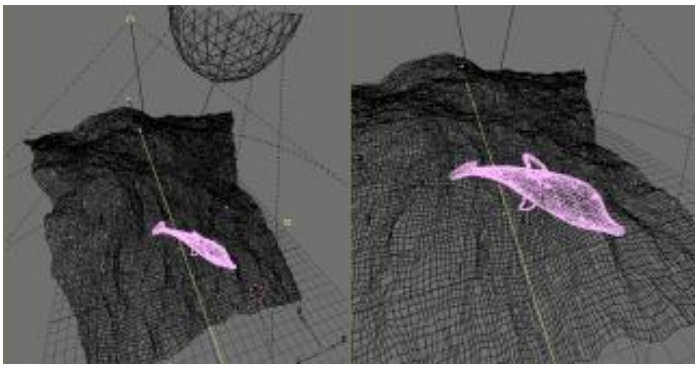
Sometimes you may get lost inside 3D space a bit. To restore the view, either click on the home icon at the bottom of the 3D window, or press Home. Blender will resize the view to include every object in your scene.



Show the entire scene.

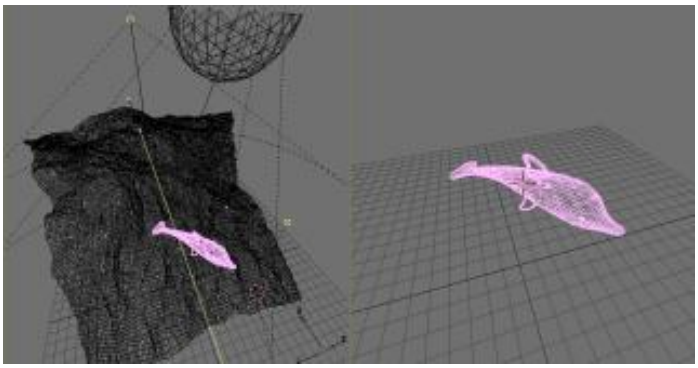
The Blender User Interface components work in the same way as the 3D Window - you can translate and scale any component. The Home key works, too.

If you want to get a specific object into view and not the entire scene, first select the object by right clicking on it (the object will turn pink to indicate it is selected). Next, press numpad dot.



Zoom in on the selected object.

A similar function like the previous one, you can also zoom in to the selected object and hide all other objects at the same time by pressing numpad /. To unhide your objects, press / again.



Showing only the selected object.

www.blender.nl
this document is online at http://www.blender.nl/showitem?id=47

Tutorials
2000 10 18

Saving

Willem Zwarthoed

id47

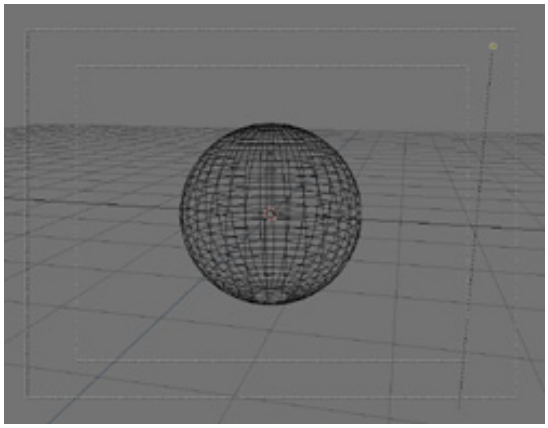
Introduction

Saving stills and animations in Blender is essential to showcase your work. Since a lot of questions are asked about this I've prepared this little mini-tutorial on how to save your work.

Saving stills

Before you can save an image you need a scene to render, so let's set up a little scene.

Start Blender and add a UV Sphere. Make sure you can see the sphere in camera view. Also add a lamp, so the sphere will show up in the rendering.



The size buttons

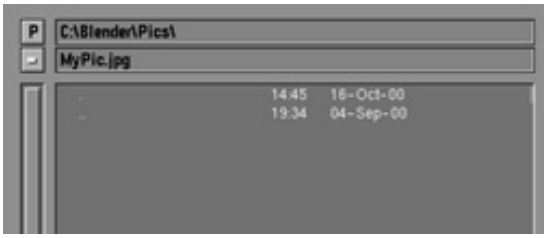
Before we render the image, set the size of the image with SizeX, SizeY and the percentage buttons located in the RenderButtons F10. Now render the image with F12 and hide the render window with F11 or Esc.

TIP: use the percentage buttons to quickly do low resolution test-renders while you're working.

The filetype buttons

Now we need to select the filetype in which the image will be saved. Locate the filetype buttons on the right of the RenderButtons and select the filetype for your image.

Note: stills can only be saved as Targa or Jpeg.



The file window

Now press F3 to save your image. A file window will open in which you can select the directory the image should be saved in. Type the name of the directory in the first and the name of your image in the second text field and press Enter twice to save the image (the first time is to confirm the file name; the second time actually saves the file).

Note: you have to include the file extension (.tga or .jpg) when you name your image!

Saving animations

First we'll animate the sphere, so we'll have an animation to save. At frame 1 select the sphere, move it a little to the left of the camera view and insert a location key with I. Advance to frame 51 (press ↑ five times). Grab the sphere again, move it to the right of the camera view and insert another location key. Go back to frame 1 again and press Alt A to verify the sphere is animated. Press Esc to cancel animation mode.

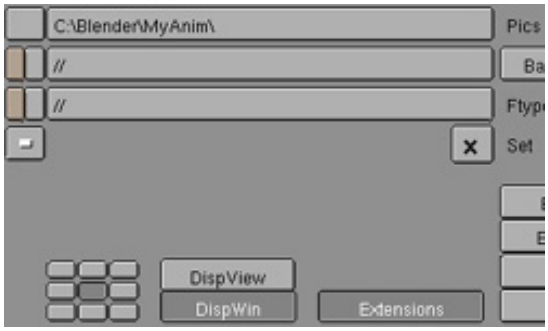
Before we press 'Anim' to render the animation we need to make a few preparations. Set 'Sta:' to 1 and 'End:' to 51. Set the filetype again. Set it to 'AVI Raw' or 'AVI jpg' to create an animation in a single file or select 'Targa' or 'JPEG' to save all the frames as separate images. Also make sure the button labeled 'Extensions' is set and don't forget to set the resolution (usually animations need to have a lower resolution than stills!).

Note: you need the Motion Jpeg codec to view AVI jpg files. Get it from here.

Blender automatically saves animations in the path and with the name indicated in the 'Pics' textbutton, located in the top left of the RenderButtons. There are a few things to consider when entering a path and a name:

Paths should end with '/' (or '\' if you're using windows). Anything after the last '/' is used to name the animation.

Blender will also assign names to animations automatically: AVI's will be saved as 'NameStartframe_Endframe.avi', individual frames will be saved as 'NameFramenumber.tga' (or .jpg).



The animation's directory & name

For example: enter 'C:\Blender\MyAnim\' to save the animation as '0001_0051.avi' in C:\Blender\MyAnim\. Enter 'C:\Blender\MyAnim' (without the last '\') to save the animation as 'MyAnim0001_0051.avi' in C:\Blender\.

The last thing to check is that there are no spaces in the Pics textbutton. If there *are* spaces Blender will generate an error: 'ERROR: Open Movie' and your animation will not be saved. Blender will also generate this error if the specified path doesn't exist.

After you've entered the path and name hit 'Anim' and your animation will be saved.

Done!

Well, that concludes my mini-tutorial. I hope this has cleared up any problems you might have had concerning the saving of stills and animations, but if there are any questions or comments please send an e-mail to support@blender.nl.

Happy blending and now that you know how to save your work feel free to show us what you've made!

Zycho

Feedback

MatrixNAN 2000 10 19

Could you write into your tutorial large rendered files vs small rendered sizes in terms of kilobytes vs megabytes. Your example rendering are so small in your download section and I can never seem to get them that small at the same resolution as the animation in the download section.
Thanks In Advance,
Nate Nesler

B@rt 2000 10 18

Thanks - I updated the article.

ynedelin 2000 10 18

hey, good tutorial. I could not save in the beginning and was getting very frustrated. I was doing one thing wrong that you might want to add to your tutorial :) . Other people have done the same mistake . one of them actually told what i was doing wrong. I am talking about saving stills JPEG and TARGA . . If one follows your tutorial exactly step by step he/she will not the have JPEG saved ... I am referring to very last step <press ENTER> . After pressing Enter , the name of the file you are saving turns its color from white to black and if then you will try to go it to the directory where you saved the file you will not find it there. You will have to press Enter twice (one more time) so Blender returns to the 3D window. well that was longer then it should have been. hopefully helps. later. Yury

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=4

Building a Castle

Bart Veldhuizen

2000 07 19

id4

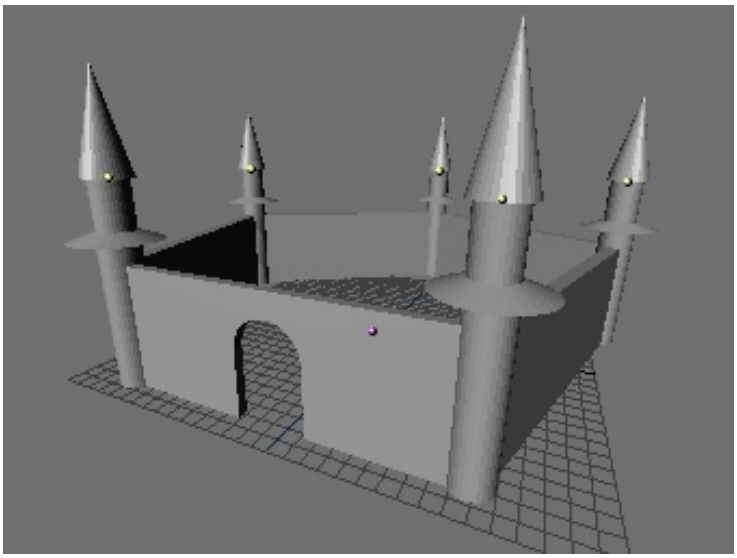
Introduction

I have always been fond of castles. I simply can't resist the urge to create one in Blender every now and then.

This tutorial will introduce you to basic modeling techniques in Blender. It assumes some basic knowledge like adding objects, using edit mode and changing the view. These are described in The Blender User Interface.

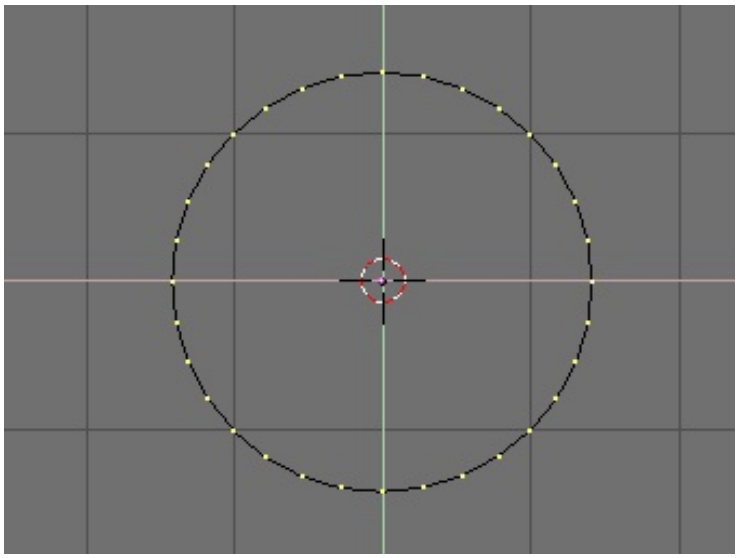
The castle is used as a starting point for other tutorials on this site like Texturing a Castle.

To begin the lesson start Blender or press Ctrl X (which starts a new project) if you already have Blender open.



The final result.

Remember to save your work early and often! This is a good computing practice which you should follow anyway but is even more important in Blender. If anywhere along the way you find that your castle doesn't look quite right you can press U and your model will revert back to the state it was in before you entered edit mode. This is Blender's version of and 'Undo' button.



A 32-sided mesh circle.

Creating the tower

The tower is created by simply extruding a mesh circle several times. By using both translation and scaling while extruding you can create a nice outline.

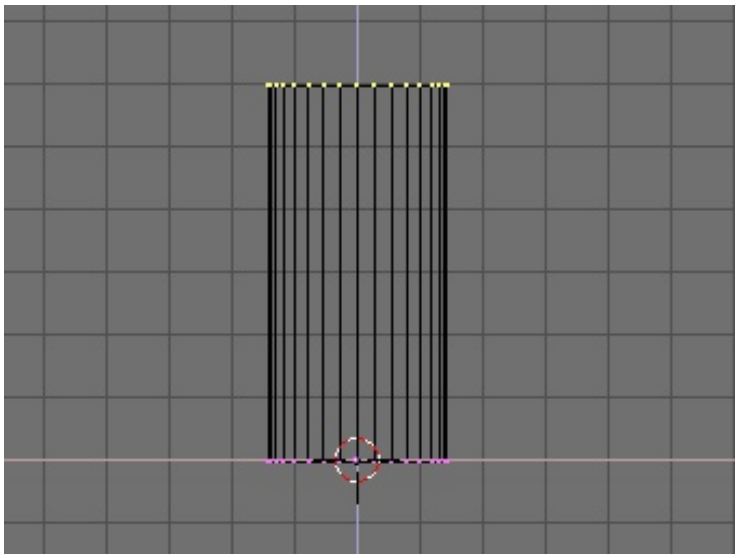
Let's start by switching into top view (NumPad 7). Remember that the each number on your keypad will switch you into a different view. Let's begin creating the top of our tower. Add a mesh circle which can easily be done by pressing the space bar, click "Mesh" and then click on "Circle." I left the number of vertices on 32 (the default) but if you are on a polygon budget, you may consider changing this to something like 10.

Make sure that you're in edit mode before proceeding.If you're in edit mode you'll be able to see the individual vertices of the circle. You'll recall that vertices are the points that define the shape of a 3D object and appear in Blender as small dots. Your screen should look like in the picture to the left at this point. If your circle just looks like a pink outline this means that you aren't in edit mode and should press the Tab key.

Switch to front view with numpad 1 and press E. Confirm the requester that pops up to start the extrusion.

Confirm the extrusion.

To confirm any requester in Blender, either click on the text (in this case, 'Extrude') or press Enter. To cancel, either move your mouse cursor away from the requester or press Esc.

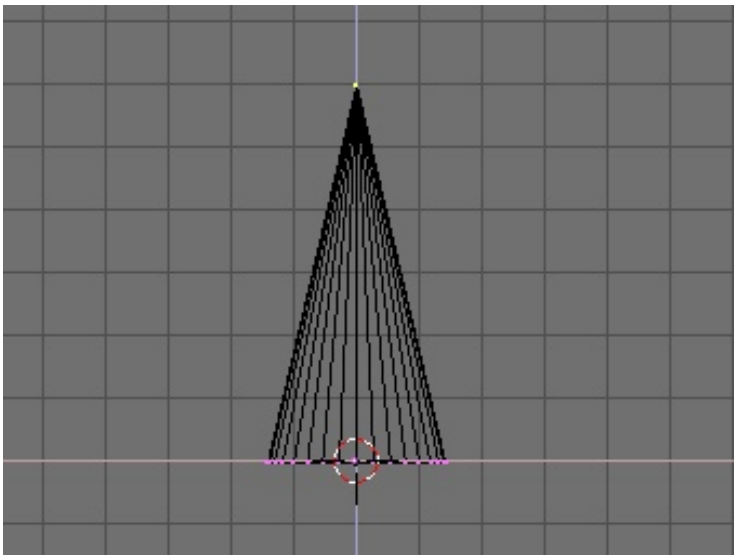

The first extrusion step

After you start an extrusion, grab mode is automatically started. Hold down Ctrl to constrain the movement to grid units. This will make your cursor snap to the grid lines in the 3D window making it easier to keep your tower from leaning to one side.

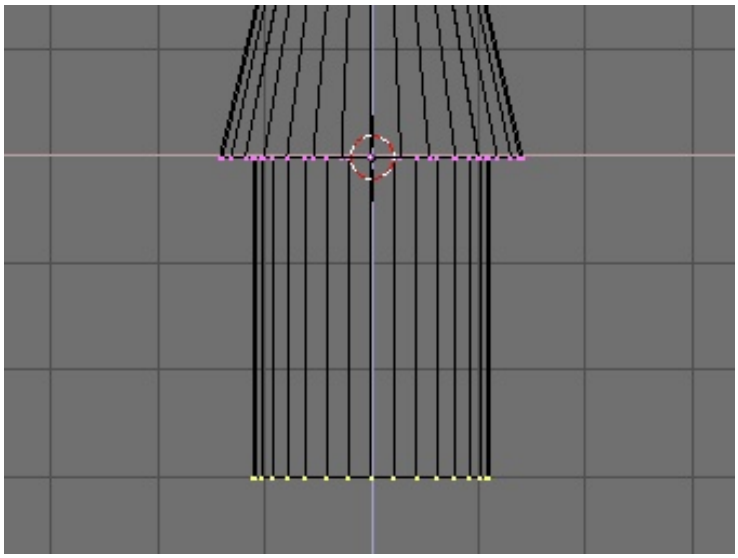
Move your mouse upwards a bit and left click. This part will form the cone on the top of the tower. As you can see extruding an object creates a clone of the vertices which you have seletected and allows you to extend the body of your object. If you had not been in edit mode during this step you would've been unable to extrude your circle.

To taper the top, press S to start scaling and hold down Ctrl again. Move your mouse horizontally to scale and left-clickonce the top has been scaled down to zero (this is shown in the bottom left corner of your 3D window).

Now I will work from the bottom of the cone and extrude the rest of the tower out of it - this uses the same technique as the previous step. First, press A to deselect all vertices (pressing it again will select all the vertices). Select the vertices on the bottom of the cone by starting Box Select (B) and dragging a rectangle around them.


Tapering the top.

Press B twice to use Brush select - in this mode, a circle appears. Use this circle to 'draw over' the vertices and select them. Press numpad + and numpad - to change the brush size.

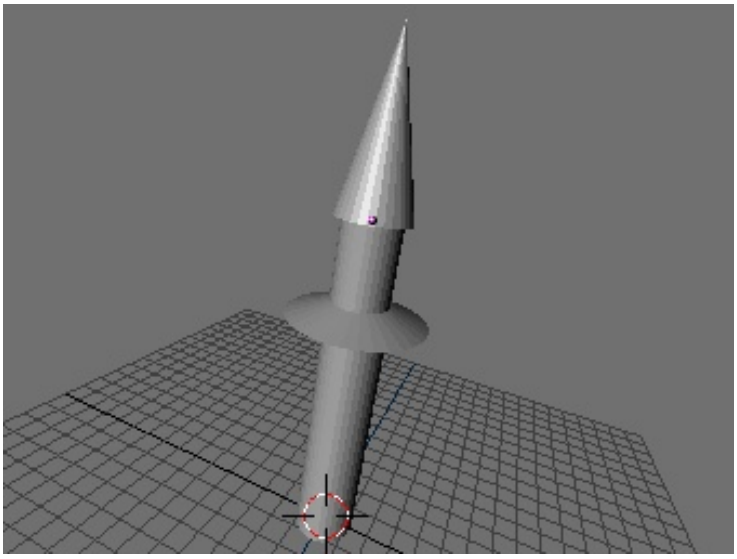


Some extrusion steps.

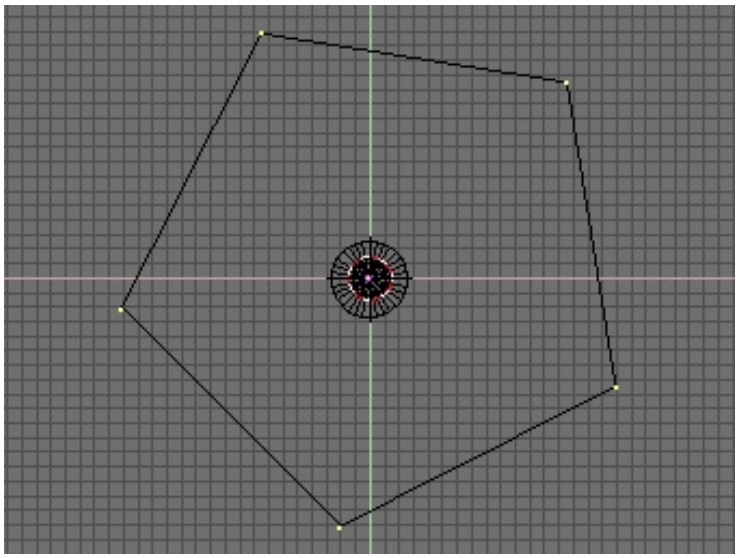
Now we're going to create the body of our tower. For the first step, start the extrusion (E key) and then immediately press S. Scale the inside of the tower down a bit. When you're satisfied with the width of the body of your tower do a left click to set the size. Next, do a 'normal' extrude and drag the selection down.

Don't forget you can use the Ctrl key to help keep your tower standing straight and tall. Left click when you're happy with the height of your tower to lock it's size in place.

If you like, you can add some more detail like I have done here. Your tower is ready!



A tower.



5-sided circle.

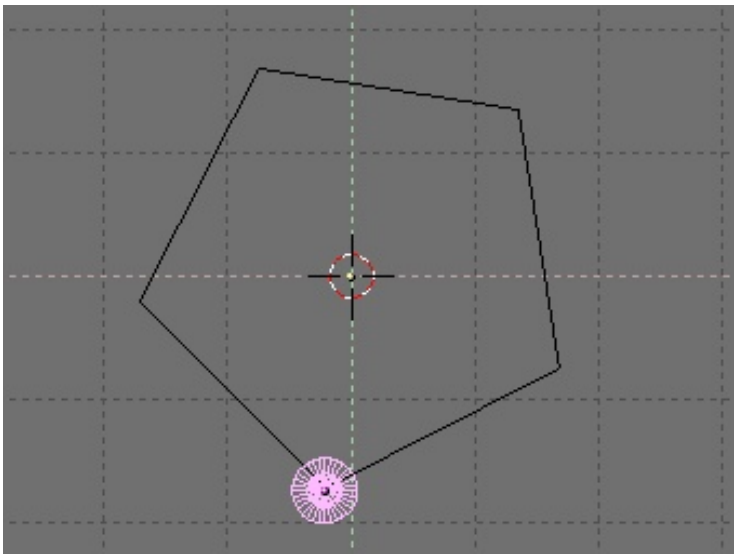
The Castle Wall

I want to create a pentagonal castle wall. To be able to place the towers on the right positions, I use another mesh circle. Later on, I turn this circle into the castle wall.

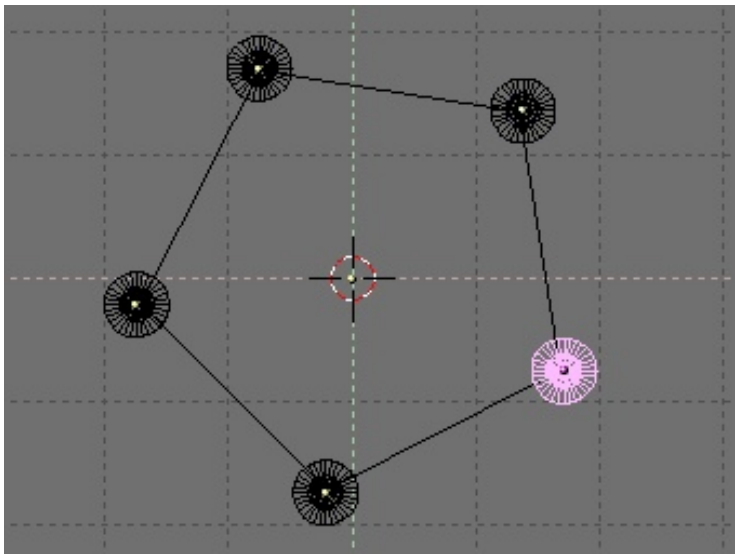
Leave edit mode (Tab), return to top view (NumPad 7) and add a mesh circle with 5 sides. Scale it to the right proportions.

By leaving Edit Mode and *then* creating our walls (a 5 sided circle in this case) we're telling Blender to think of our circle as a seperate object from the tower. This will make editing our castle wall a lot easier later on.

Leave edit mode and select the tower with . 'Grab' your tower (key) and move it to one of the corners of the pentagon. Don't put the center of the tower exactly over the corner of the pentagon, but rather slightly more outward - this is to take the thickness of the wall into account (see below).



Placing the Tower.



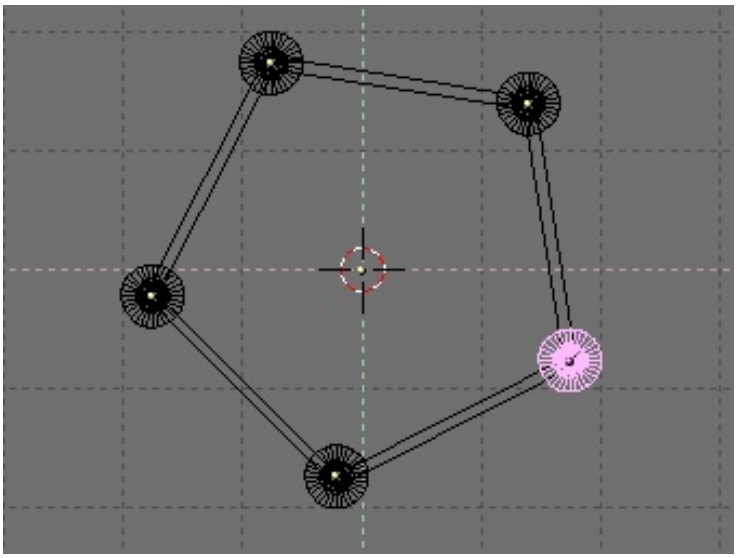
Creating copies of the tower.

With the tower selected, press Shift D to create a duplicate. Grab mode is now automatically started, so you can move the new tower to another corner. Repeat this for all corners.

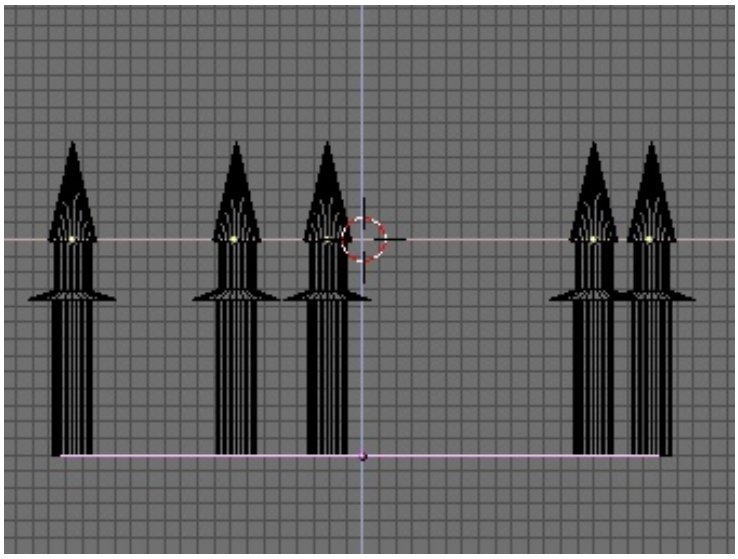
Pressing Shift D will make a full copy of the entire object - geometry included. After this, you can edit the geometry of every individual copy without affecting the others.

If you use Alt D however, Blender will create an instanced copy. In this case, each copy's geometry is linked to the original data. Changing one individual copy will change every copy along with it. Using instanced copies requires less memory than full copies.

Now to give the castle wall some body, select the circle and enter edit mode. Select all vertices and start extrusion. Scale the selection a bit to give the circle some width.



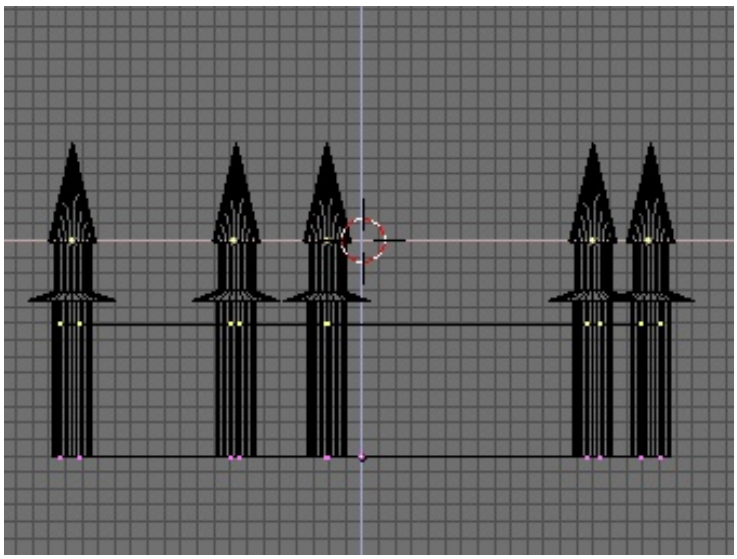
Giving the wall body.



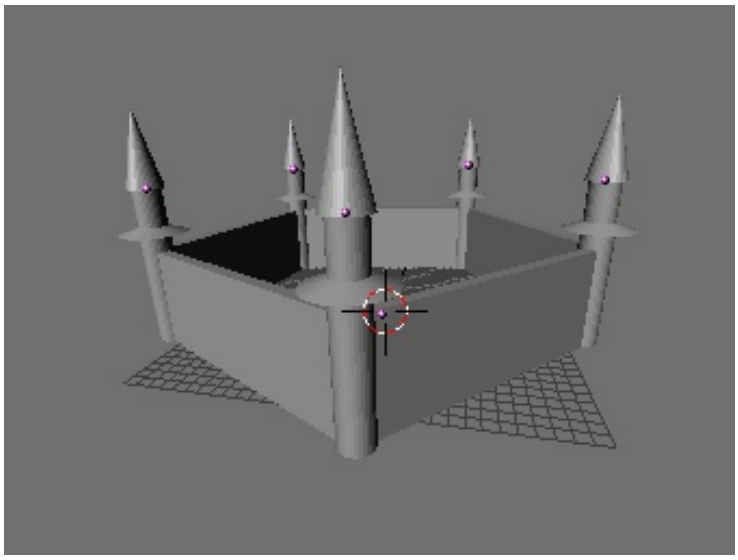
Giving the wall height.

The wall has width now, but no height. Leave edit mode and switch to front view with numpad 1. Your wall now probably starts somewhere in the middle of your tower, so use grab mode to move it to floor level.

While still in front view, extrude the wall upward.



Giving the wall height.



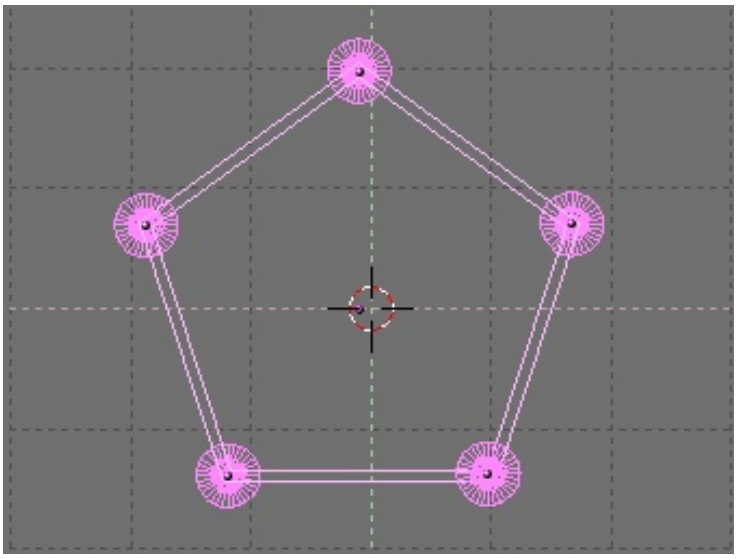
The result so far!

This is already starting to look like a castle, isn't it?

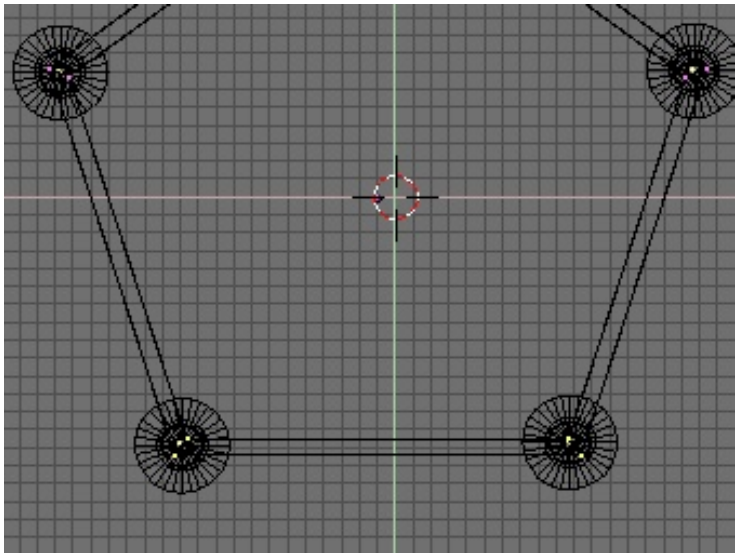
Creating the Castle Door

This is all very nice, but there is no door in the wall yet! Instead of using a Boolean operation to cut it right out off the wall, I will show you how to do this with simple mesh editing.

The segment of the wall that I want to work on (the bottom part) is slightly rotated, so I will first make it horizontal. Select all the objects in your scene with A and rotate your entire scene until the bottom segment is horizontal. Hold down Ctrl to constrain the rotation to multiples of 5 degrees.



Rotating the scene.



Selecting the bottom wall.

Now I separate the segment of the wall from the rest so that I can work on it. Select the wall and enter edit mode. Use Box Select to select the bottom segment of the wall as shown.

To create a new object out of the selected vertices, press P and confirm the requester. If you now leave edit mode, you will have a new, separate object. Select the new object.

Creating a new object out of the selection.

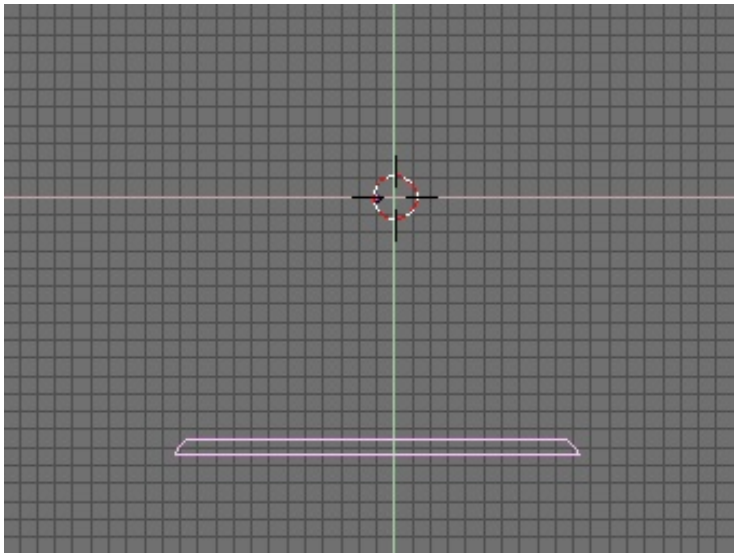
Moving the wall segment to another layer.

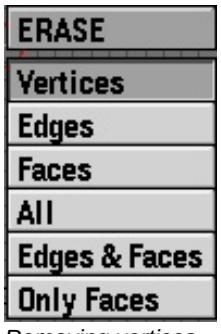
Now to make life a little easier on myself, it is a good idea to move this new object into a separate layer. Doing this will allow me to hide the other objects. With the new object selected, press M to bring up the Move Layer requester. Select the second button or press 2 to move the wall segment to layer 2. Press Enter when done.

In the bottom of your 3D window, you see a similar row of buttons. It currently has only layer one selected, so moving the wall segment to layer 2 will make it disappear. Shift-click on button two or press Shift 2 to activate layer 2 as well.

It's a good idea to utilize layers for anything more than a fairly simple project. This will allow you to break down your project into more manageable and flexible chunks. It will also enable you to make changes far more easily than if you had everything on one jumbled, complicated layer.

For now, I only want to see the wall segment so press 2.

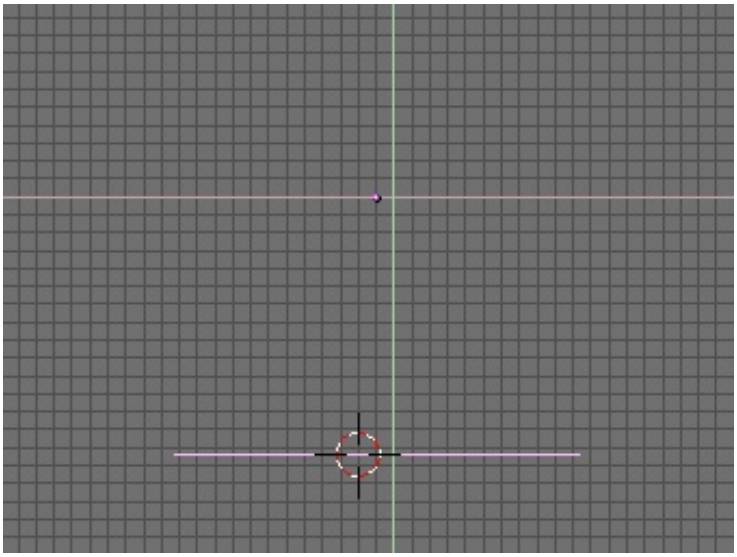

Showing only the wall segment.



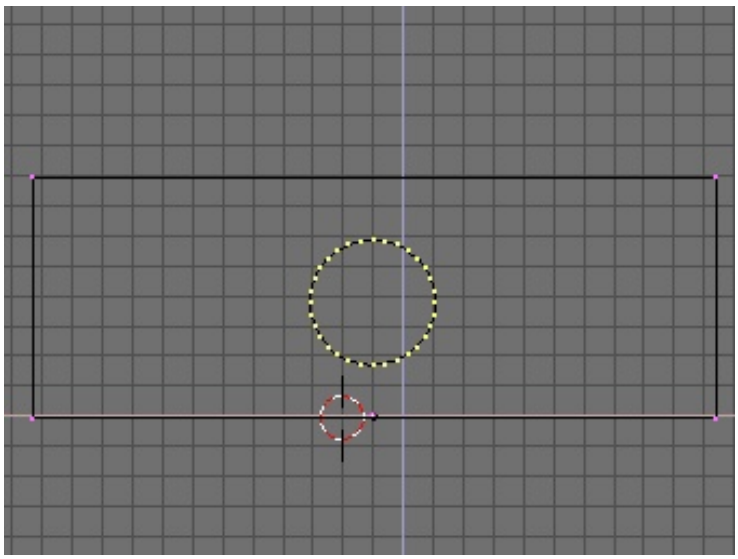
Removing vertices.

First I will create a flat image of a wall with a door in it - later I can extrude this to get a wall with a hole in it. First I need to remove half of the vertices. Enter edit mode and select the top vertices using Box Select. Press X to bring up the Erase requester. Select 'Vertices'.

In top view, carefully place the 3D cursor on the line. This determines where the next shape will be placed. If you don't line up your 3D cursor on this line your door won't come out correctly later!



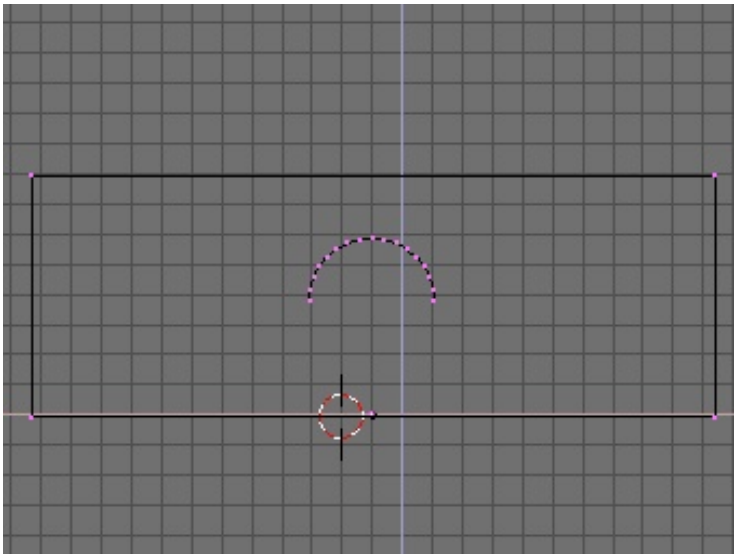
Placing the 3D cursor.



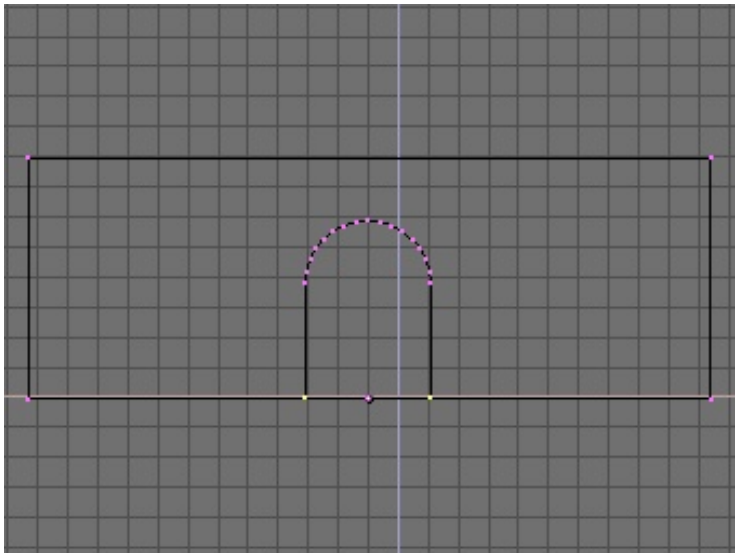
Adding a circle to the door.

Switch to front view, enter edit mode (!) and add a 32 segment mesh circle. This will become the top of the door. Move and scale it a bit so that it has the right size and position.

Now press A to deselect all vertices and select only the bottom half using Box Select. Press X and select 'Vertices' when the Erase requester pops up to delete the bottom half of the circle.



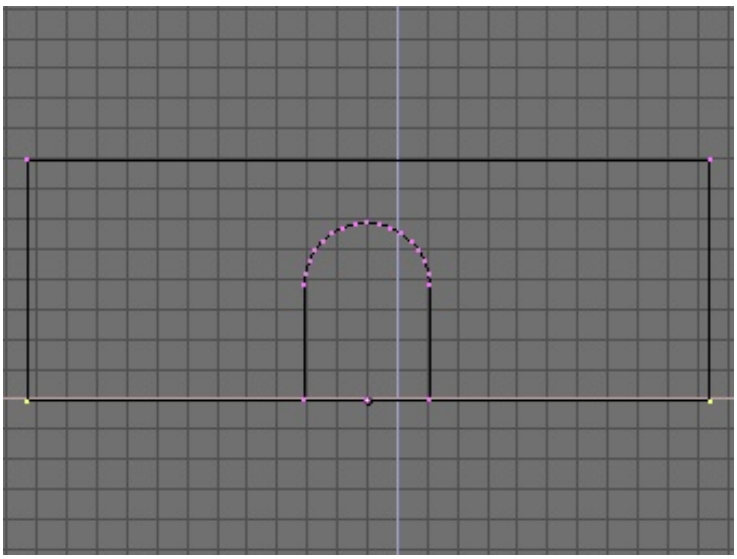
Deleting half of the circle.



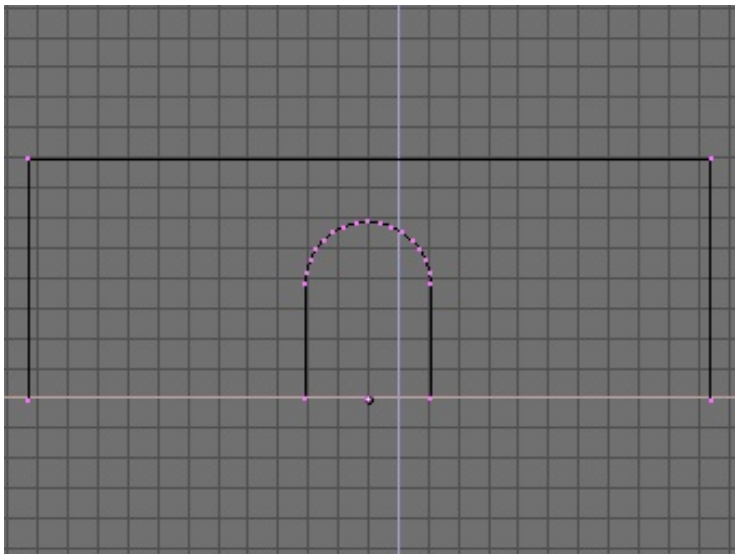
Extruding the circle to form a door.

Select the two lowest vertices and extrude them downwards so that they match the bottom edge of the castle wall.

Now I need to attach the door to the rest of the wall. First, I need to remove the original edge at the bottom of the wall. Select the two vertices at the bottom left and bottom right corners of the wall as pictured on the right.



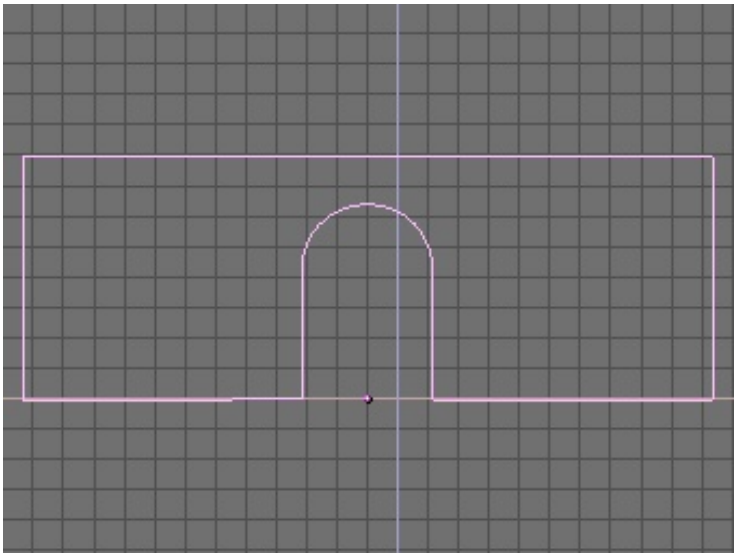
Preparing to remove the bottom edge.



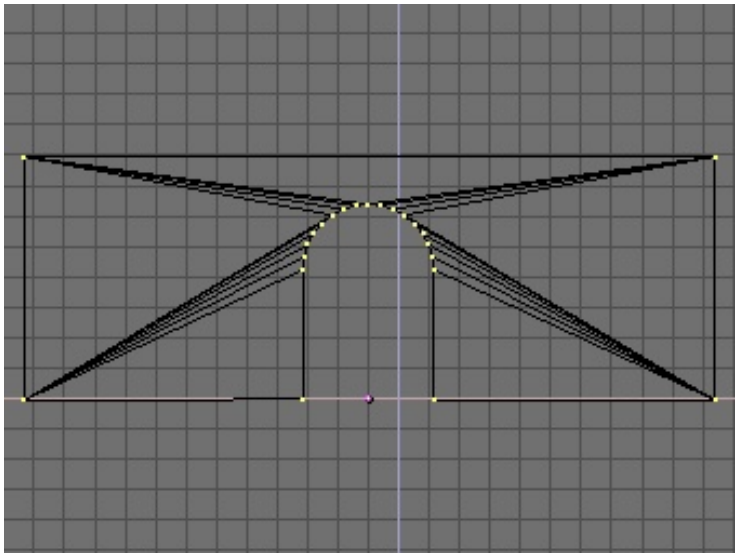
The bottom edge removed.

Press x and select 'Edges' to remove the bottom edge.

Connect the corners of the wall to the corners of the door by selecting two vertices and pressing F. If you leave edit mode now, your model will look like this:



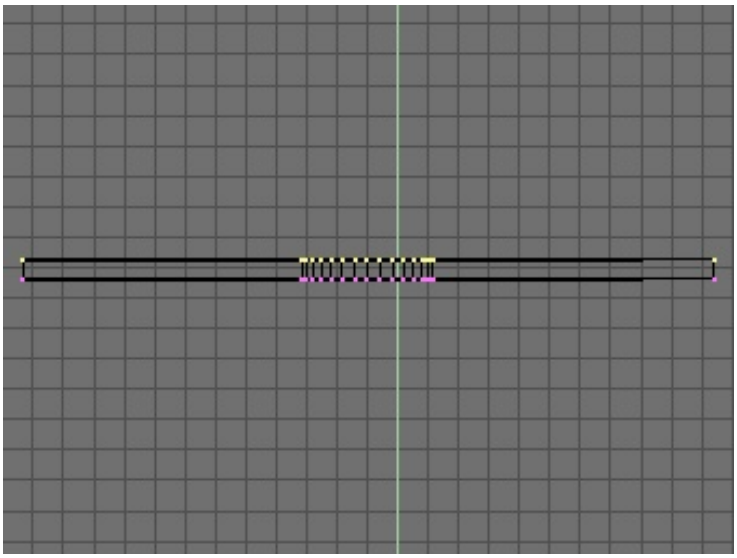
Reconnecting the outline.



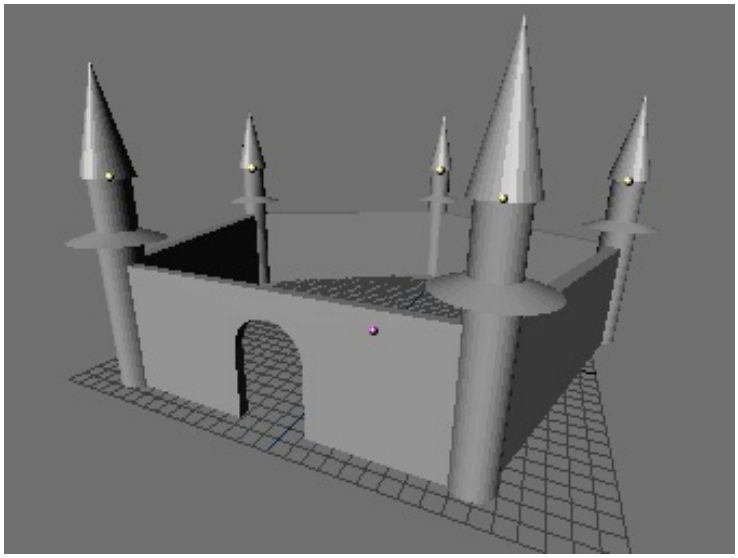
Filling the wall.

The outline is now all right, but the shaped is not filled. You will notice this when you switch to shaded view with Z. To fix this, enter edit mode and select all vertices. Press Shift F to fill the shape.

As the last step I will give the wall depth again. Switch to top view and extrude the wall slightly upwards. Try to match the depth of the other wall segments. If you are not sure about the depth, press Shift 1 to turn on layer one.



Extruding the wall and door.



When you are done, leave edit mode, activate layer 1 (Shift 1) and turn on solid view with Z. I also turned to perspective view (Numpad 5) and rotated the view a bit.

Quite impressive for only a few steps, isn't it?

The next step is to apply some nice materials to your model. This is explained in detail in Texturing a Castle.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=5

Tutorials
2000 07 18

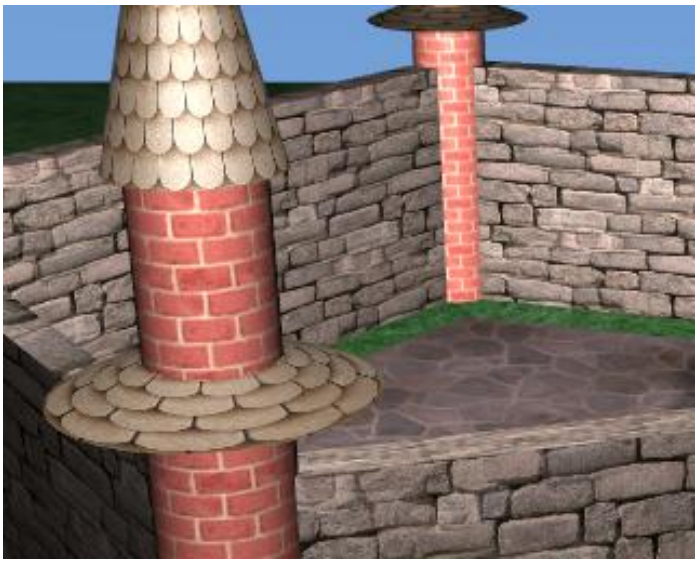
Texturing a castle

Bart Veldhuizen

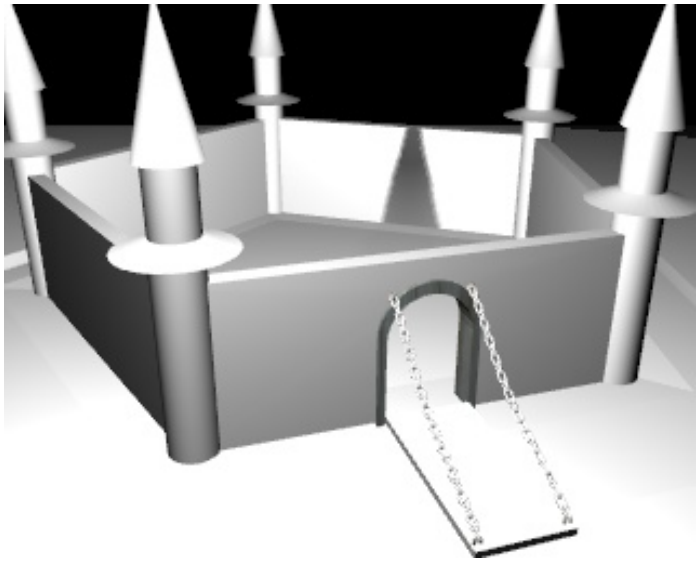
id5

Introduction

No matter how well you can model, you will need to add good materials and textures to your model to make it look realistic. In this tutorial you will learn how Blender works with materials. It is based on Building a Castle.



The final result.



The castle before applying materials.

You can either do that tutorial first, or download one that I made below. You will notice that I added some details like a drawbridge and a moat. All of these were made with the same techniques as described in Building a Castle.

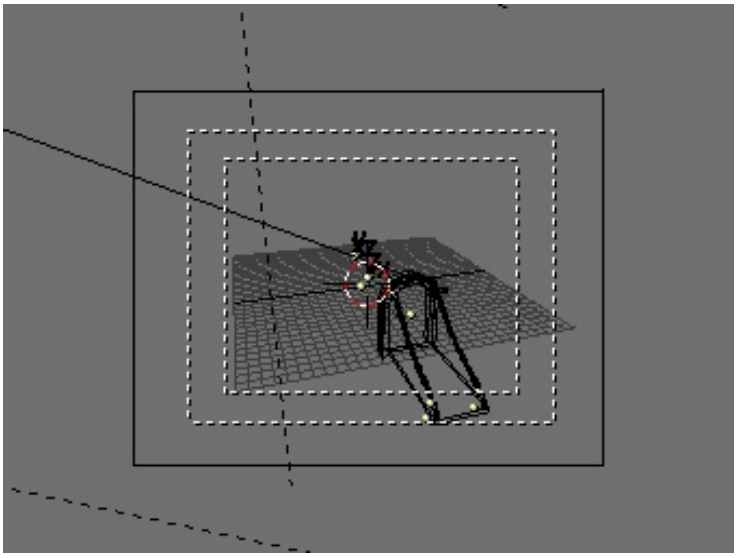
It also contains two lamps and a camera.

Download: castle.zip

To make things a little easier for you, I have moved the objects in my file to different layers. This allows you to display only relevant objects by the push of a button.

1 - Moat
2 - Floor
3 - 1 Tower (the rest is in layer 9)
4 - Front wall
5 - Door, drawbridge and chains
6 - Inside square
7 - Water
8 - Other walls
9 - Other towers

To display a layer, press the associated number on your keyboard. To show all layers at the same time, press the ~ key. As you can see, the light and camera are present in all layers so you can always render your objects.



Showing only the drawbridge in layer 5.

Switching to the Material Buttons screen.

Creating a new Material

Let's start off by creating a simple material without textures. Select the moat object in layer one by pressing 1 and select the object.

To start working with materials, go to the Material Buttons screen. You can reach this by selecting the icon of the red ball in the header of the lowest window, or by pressing F5.

Later on I will add a transparent water surface layer on top of this.

Right now, there is no material associated with the object so nothing is shown in the Material Buttons window. The first thing I need to do is add a new material. Take another look at the header of the Material Buttons window and press the button with the '-' sign on it.

The material data block button.

Selecting an existing material or creating a new one.

A requester will pop up, asking you if you want to select an existing material or create a new one. In this case, select 'Add New'.

Let's take a closer look some of the buttons in the window that appears now. On the far left is the material preview. Any changes that you make to your new material are immediately shown here. The default view is flat but if you prefer a sphere, just press the icon of the red ball.

The material preview

Colors

The next group of buttons determines the material color. Try changing the values of the R, G and B sliders now to create a nice deep blue color for your moat. The changes will be visible in the 'Color' section of this window in real-time. If you stop dragging a slider, the material preview will render with your selected color. I used R=0.1, G=0.3 and B=0.6 for my material.

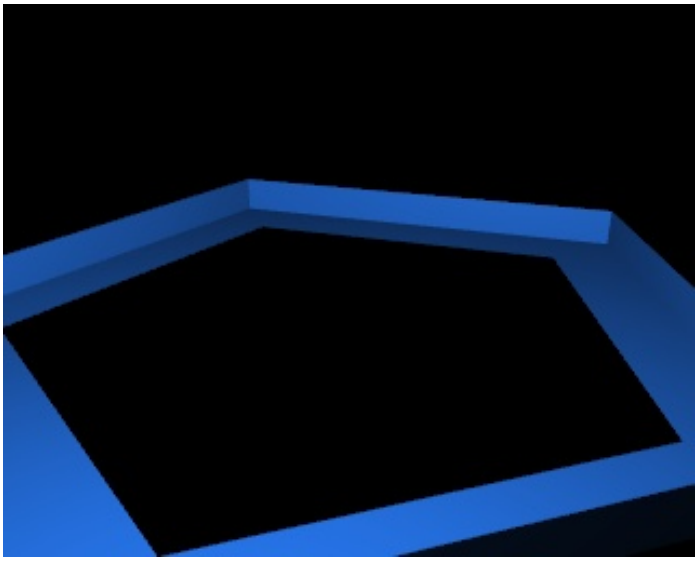
These buttons affect material properties such as shininess, hardness and transparency. Try changing them and seeing how they affect the look of the material preview (this works best of you view it as a sphere instead of a flat plane). I changed the Spec value to 1 and Hard to 75 to get a nice plastic.

Material properties

Changing the material name

When you are satisfied with your new material you need to give it another name. This makes it easier if you want to apply it to another object later. Left-click in the material name space (it probably says 'MA:Material') and type in a new name. Press Enter when done.

Now let's take a look at your first material! Press F12 to render (if you haven't downloaded my castle you may need to add some lamps. If you don't have any lamps your render will turn out all black!!).



Blue!

Manipulated castle wall picture

I could go on adding materials like this, but the scene would start to look like everything was made out of plastic. To add some realism I will now apply an image as a texture map. For this, I have prepared a picture of a castle wall.

Download: wall_seamless.jpg

You will see that I have manipulated it a bit to prevent the edges of the picture from showing up. If you are familiar with the Gimp (a free image manipulation program), you can do this with the feature 'make seamless'. Preparing your images before you apply them as a texture map is often a good idea.

The 8 texture channels.

Switch to layer 4 and select the front wall. Create a new material for it. Now take a look at the right part of the Material Buttons screen. There is a row of 8 buttons. Each one can hold a texture that you can apply to a material. Click on the first button and add a new texture by selecting the '-' button below the texture buttons. This button works in the same way as when creating a new material.

The Material data button

Like with the material, you can give each texture a name by left-clicking in the name field. If you want to remove a material, select the 'Clear' button.

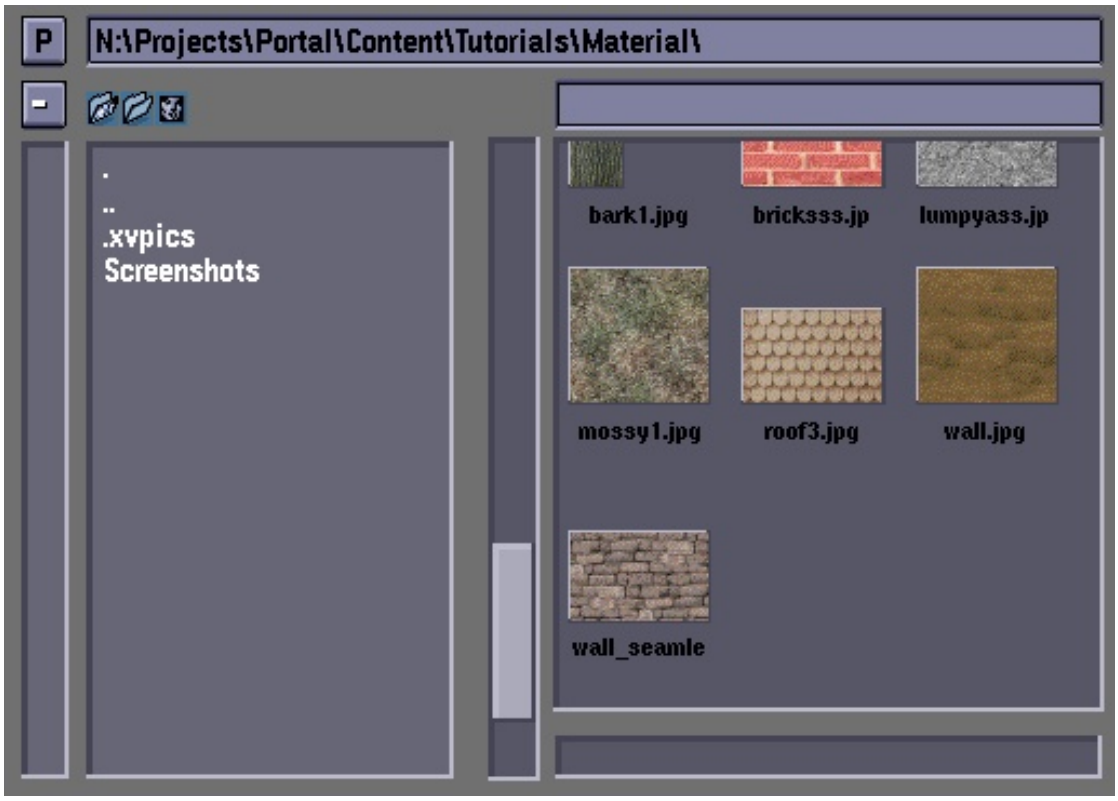
To change the texture settings, switch to the Texture Buttons screen. Select the texture icon or press F6.

Switching to the Texture Buttons screen.

Preparing to load a picture.

For now, it is best to ignore all the buttons in this screen except for one: 'Load Image'. Select it to bring up the Select Image window.

This window works like any other file window in Blender, except for the fact that it shows thumbnail previews of each image in the current directory. If you have many images, it can take a while for the screen to update. To load your image, select it by left-clicking on it and pressing Enter. Alternatively, you can also middle-click on it to immediately load it.



Selecting a texture map

Determining the mapping method.

Return to the Material Buttons screen and take a look at the mapping method buttons. These determine how the texture is applied to your object. The most basic one is 'Flat'; this works like a slide projector. If you are not careful with this one it can severely distort your image.

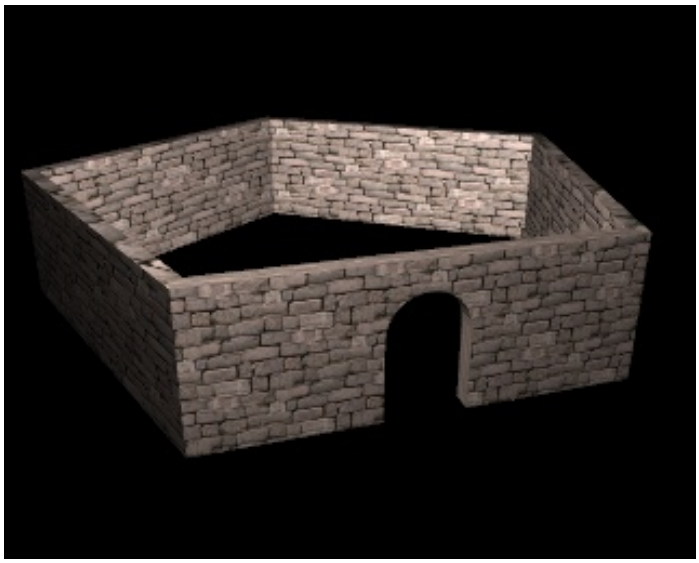
The second option, 'Cube', projects your map on all sides of your object. In many cases, this works quite well without further tweaking.

'Tube' and 'Sphere' are, you guessed it, for mapping your object onto tubes and spherical objects.

For the wall, select Cube mapping.

If you do a test render now (F12), you will see that the bricks are still quite large in comparison to the wall. This is where the Size buttons come in handy: tweak these to change the size of your bricks. You can squeeze your image by scaling your texture map differently in the X or Y direction. I found that SizeX=3 and SizeY=1.5 gave good results. Your mileage may vary though if you are using your own wall with a different size.

Changing the mapping size.



The wall with a brick map.

So far, so good. Now on for the rest of the walls. Press Shift 8 to turn on the other walls (Shift will keep layer 4 selected as well) and select the other walls. Use the material selection button ('-') to apply the wall material to this object too. A test render should look look like a real wall now!

Mapping the tower

Mapping the tower is very similar to the wall, only this time I will use a different mapping method. Because of the circular geometry of the tower, it makes no sense to use cube mapping for it. Instead, I will use cylindrical mapping ('Tube') for it. The the rest, the tweaking is the same as above.

The roof tiles.

The tower consists of two separate objects so start by creating two new materials for it (Roof and Wall). Load the provided textures into them.

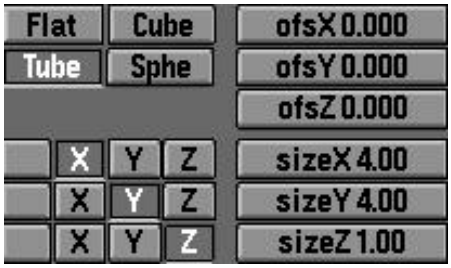
Download: roof.jpg

If you are using a tower that you modeled yourself, you will need to split the roof from the rest of the tower. To do this, enter editmode and box select the vertices that you want to separate from the rest of the object. Press P to separate them.

This is the texture that I used for the wall of the tower.

Download: bricks.jpg

The tower wall.



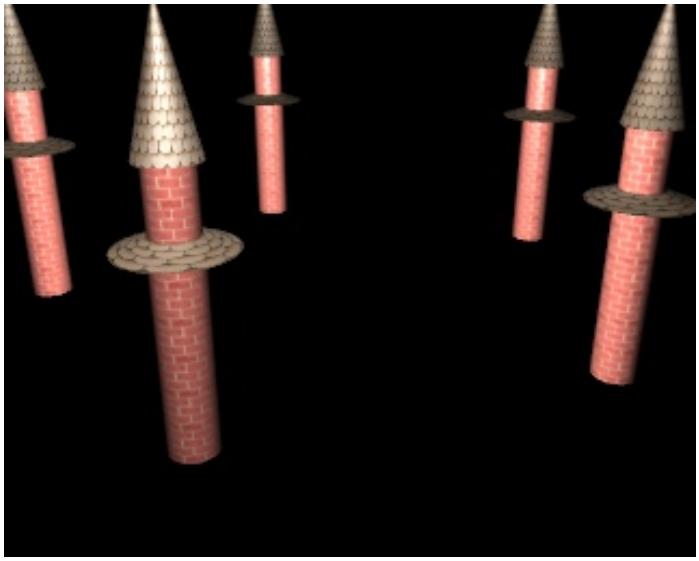
Setting the tower mapping parameters.

Again, you will need to tweak the size of the applied texture map before it looks all right. I had to create a second material for the roof halfway down the tower and tweak it's size settings a bit before it looked ok.

I found that my brick texture looked a little bit too fuzzy. This was caused by the way Blender handles texture maps - each texture map is filtered before it is applied to an object. In this case, the filtering was a bit too much. Fortunately, you can change the amount of filtering that is applied. Go to the Texture Buttons screen and decrease the 'Filter' value.

Blender can remember two versions of rendered images. You can flip between them with J. If you want to see the result of the new filter value, render an image with the old setting first. Then press J and render a new image with the new setting. Pressing J will now flip back and forth between the old and the new image.

This is a quick test render of the mapped towers.



The mapped towers.

The remaining materials

The remaining materials all use the techniques that I have described above. Just load the images provided here and try if you can figure it out on your own. If you can't work it out, take a look at the supplied .blend file at the end of this tutorial.

The Portal:

For the portal you will need to create three materials: one for the door itself, one for the wooden bridge and one for the metal chains. For the door, I have selected a sandstone-like texture. Apply it to the portal outline with a low specularity and a low hardness.

Portal texture.

Download: wall.jpg

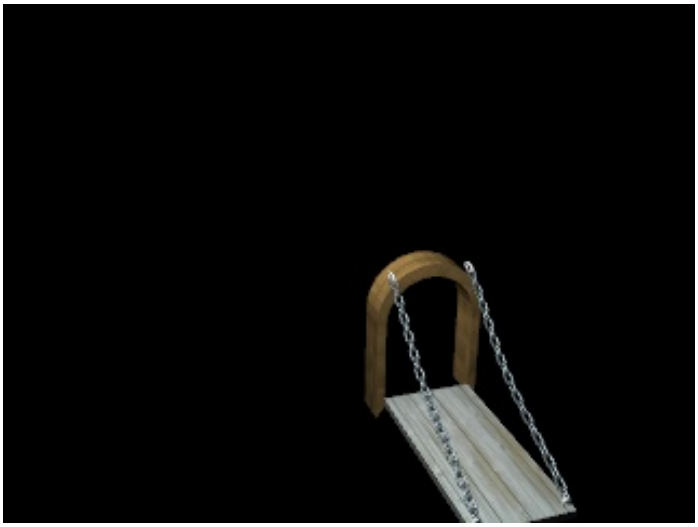
Apply this wood texture to the bridge. Like with the portal, create a dull looking surface for this.

Bridge texture.

Download: bridge.jpg

The metal chains are relatively small and do not really need a texture map; the detail would probably be lost anyway. However, to make a material metallic there is a nice trick that you should know: when you look at the real-time color preview you will notice there is also a color preview which is labeled 'Spec'. This is the specular or highlight color of the material. If you select the 'Spec' button, you can change this color.

For plastics, the specular color is white. For metals it is usually different from the material's normal color. To create realistic chains, try setting the material color to black and the specular color to blue.



This is the texture map for the courtyard inside the castle. Try experimenting with the 'Nor' button in the texture types buttons (this turns on bumpmapping). The amount of bumpmapping can be controlled with the 'Nor' slider.

Download: courtyard_seamless.jpg

Castle courtyard texture.

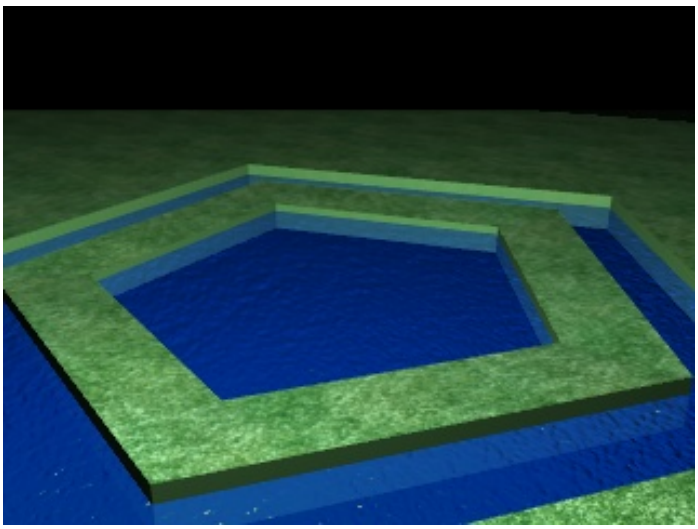
For the grass and the water I have re-used this texture. To create good looking grass, create a dull green material and apply this texture. Next, look at the buttons in the bottom-right of the Material Buttons screen. The value of 'Col' determines how much of the texture's color is mixed with the material's own color. Decrease this value to make the grass look greener while still retaining the texture's noisy character. I had to increase SizeX and SizeY to 15 to get a satisfactory result.

Download: grass.jpg

Grass texture

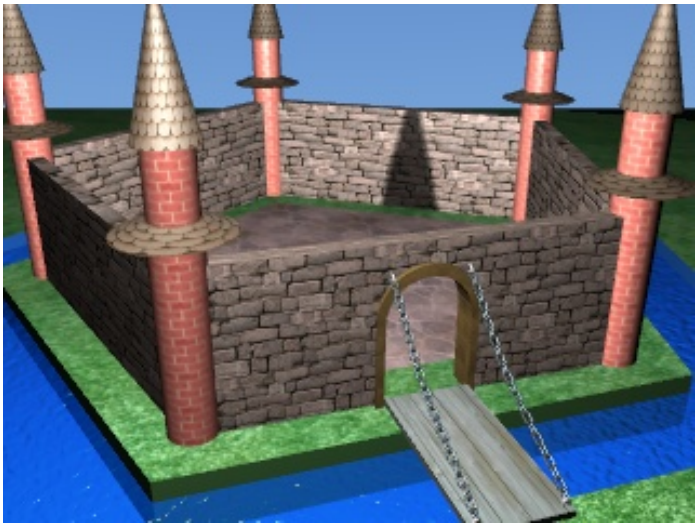
To use this texture with water, first create a blue-plastic material. Use a high specularity and hardness. Now apply the grass texture, but turn off the 'Col' button on the top right in the material buttons screen. Instead, select 'Nor'. This will turn the map into a bump-map, simulating bumps in the material.

Finish the water by setting 'Alpha' to 0.5 - this creates a transparent material. In the row of buttons in the middle of the screen, select 'ZTransp'. This option will force Blender to calculate objects underneath the water surface, too.



Grass and water

Here is the final result - quite a lot better than what we started off with, don't you think? Of course it is very important to build a good collection of textures for yourself. Take the time to clean them up, make them seamless and ready for use.



A fully textured castle.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=142

Tutorials

Append

Willem Zwarthoed

2001 01 31

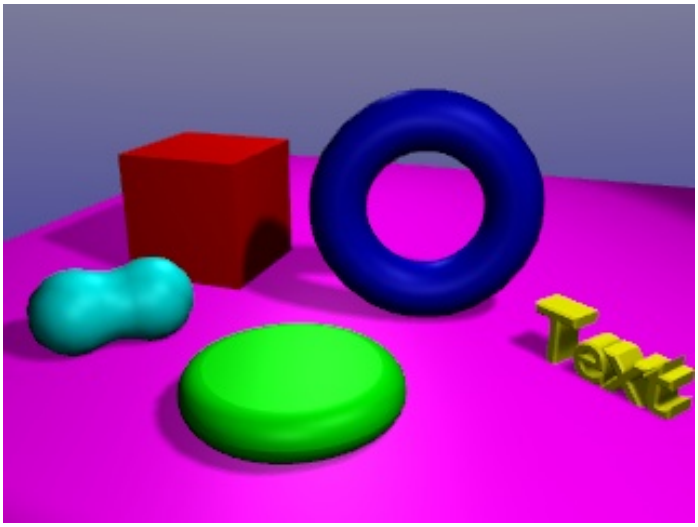
id142

Introduction

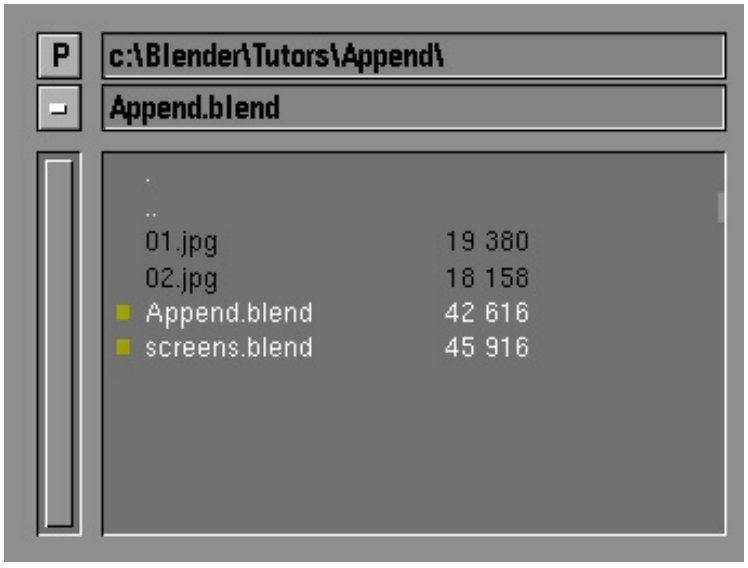
If you've been Blendering for a while you have probably run in to the problem that you're working on this cool animation, but what you'd really want is that one model you created three weeks ago. You have the model, spent an entire weekend on making it *just* right, and now you want to use it again without having to model it all over again. Here's a little 'how to' on **appending from other .blend files**.

To use the append feature you need, of course, at least two .blend files: one to append *to* and one to append *from* . Download my demo file below, start up Blender and load the file with F1. You'll see a bunch of objects of different types; mesh, curve, surface, text and metaball, all with different materials.

Download: Append.zip



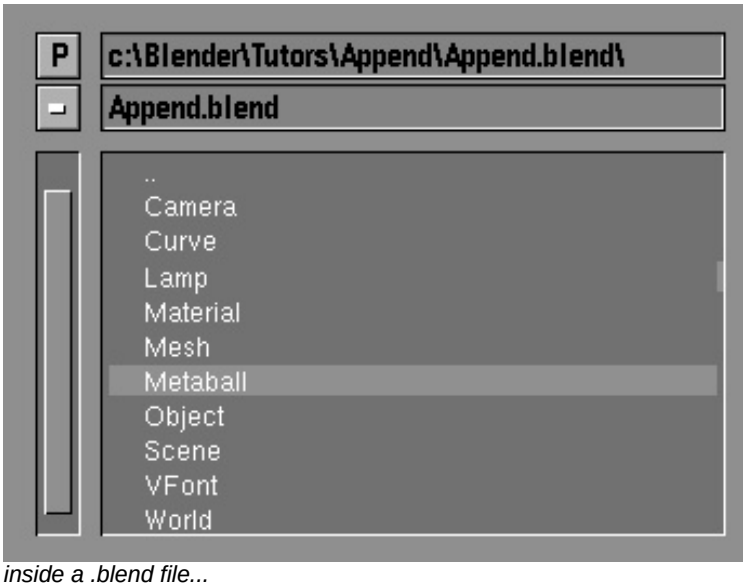
Several happy colored objects!



.blend files now show up white

Append it!

This is the file we'll be appending from. Now press Ctrl X and confirm the popup to reset Blender and delete the default plane. We'll use this setup to append to. Ok! Let's start appending models. Hit Shift F1 and browse to the demo file. You'll see all the .blend files are shown in *white*.



inside a .blend file...

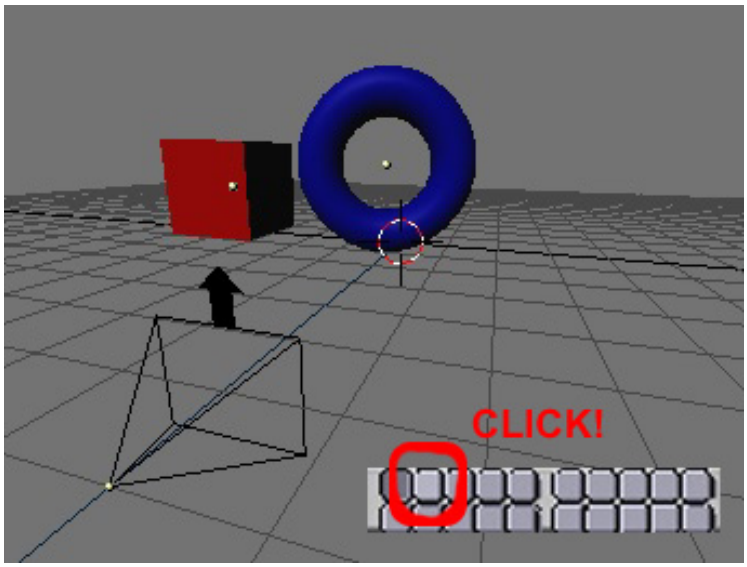
If you click the .blend file you'll see you can browse the file as if it was a directory. This way you can see the entire data structure! If you want to import models you have to look under 'Object'. Find the donut (it's called SurfDonut) and middle click it...hey presto!

In complex .blend files with many objects it is very imported to give your objects useful names! It's impossible to find an object when all your models are named Plane.001, Plane.002, ..., Plane.321 😊. You can name an object by clicking on the 'Ob:' button in the header of the EditButtons F9.

Multiple objects at once

You can also import more than one object at a time so let's do that next. Press Shift F1 again, you should still be in the Object 'directory' in the demo file. Now rightclick on Cube and CurveCircle (they will be highlighted in purple) and press enter. You should see the cube appear, but strangely enough the CurveCircle doesn't...

It is important to know that Blender remembers everything about an object when it is imported including its location and on which layer it is. If you examined the demo file close enough, you'd have noticed the CurveCircle was on layer two! So press Shift 2 or click the button for layer 2 in the 3d window header to activate the layer. As you can see Blender did import the CurveCircle!



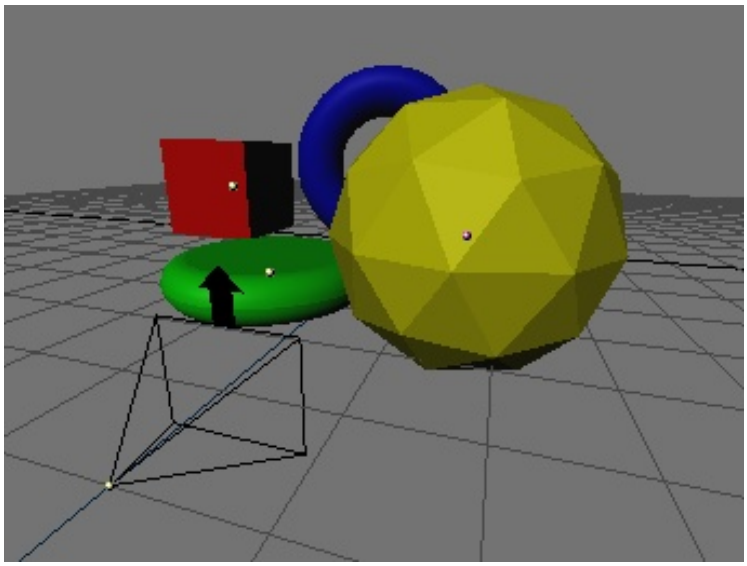
Where is that CurveCircle??

If you want to import objects with a complex parenting hierarchy you should import them all in one go. The child-parent relation will be lost if you import them one at a time!

Materials

In this way you can import any object from another file, but you can also import materials and textures this way! As a final exercise I'll show how to import a material.

Add a new object (I chose an Icosphere) and move it into clear view. I want the Icosphere to have the yellow color of the text in the demo file. Press Shift F1 and click the '..' to go one level beck in the data structure. Now click material and click yellow with (now aren't you happy I named those materials?). Nothing seems to have changed, but press F5 to go to the MaterialButtons and with the sphere still selected click the '-' button and select the yellow material we just imported (it should have a 0 in front of it, because it's not being used by any object yet).



The sphere is yellow! Yay!

And that's it. Now you know how to import any object or material from your old projects!

I hope you'll have fun with it,

Zycho

Feedback

GammaRayQ21 — 2001 01 31

thank you, this has saved me days of work.
typed it wrong the first time

GammaRayQ21 — 2001 01 31

thank you, this has svaed me days of work.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=2

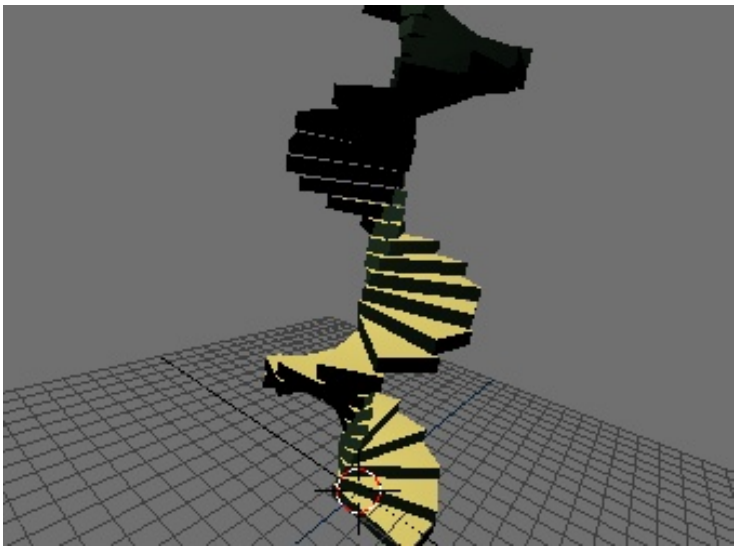
Tutorials
2000 07 18

Building a Spiral Stair

Bart Veldhuizen

id2

Introduction

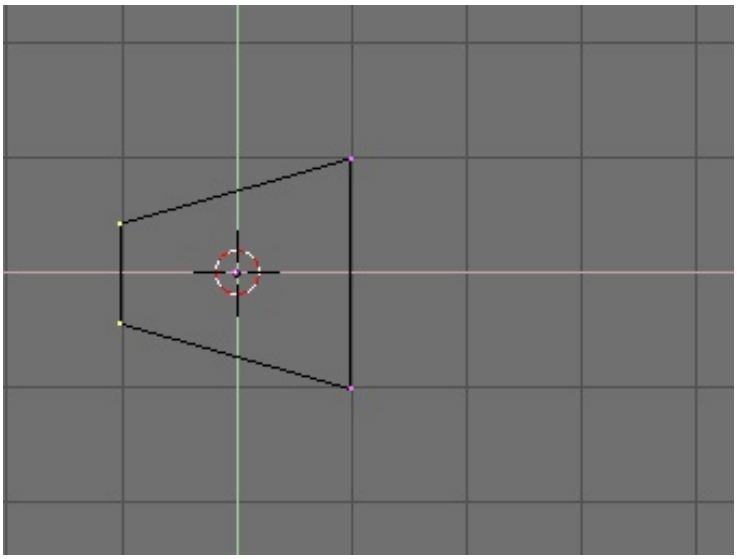


One of Blender's features is its ability to use animations as a tool for modeling. Strange as this may sound, it is often a very powerful technique.

In this tutorial I describe building a spiral stair. Using traditional methods, this would require many steps to model, but once you grasp how a moving object can represent the final result, it becomes a lot easier.

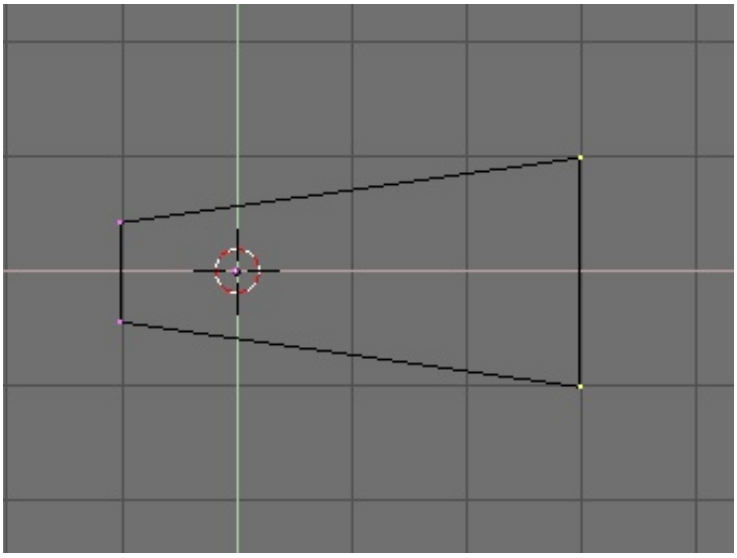
Before you start, you may want to do <Keyframe tutorial> (available soon) to learn the basics of keyframe animation.

The first thing I do is create a single step. Start with a clean Blender (press Ctrl X to clear your scene) and make sure that the default plane is selected. Press Tab to enter edit mode. Select the two vertices on the left with box select (B) and scale them down slightly (S).



Scaling the vertices at the left.

Next, select the two vertices on the right and move them further to the right (press G to enter grab mode).

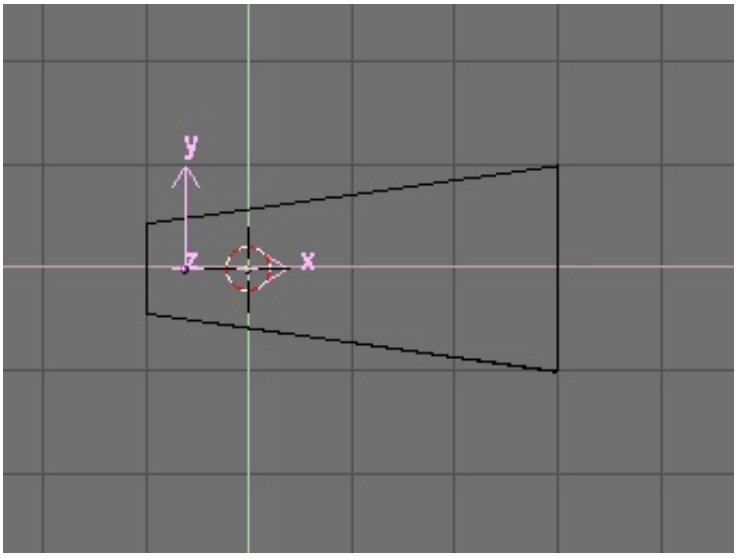


Moving the vertices at the right.

To give the step some depth, switch to front view with numpad 1, select all vertices (A) and extrude the shape slightly by pressing E and moving your mouse.

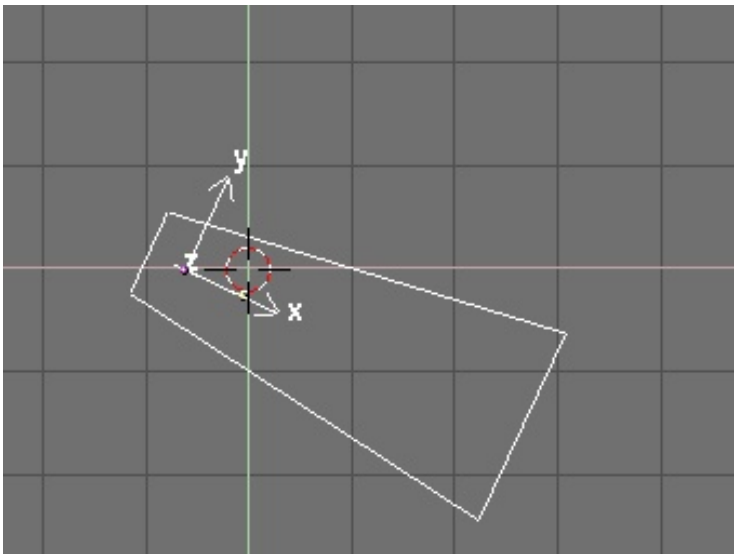
Make sure that numlock is switched on when working with the numeric keypad.

Now for some animation. To create the spiral I want to do two things: first make the step rotate around some axis, and second make it move upward while it rotates. This is made easier by using an empty as the rotation axis. Go ahead and add one now. Place it inside the step, slightly to the left.



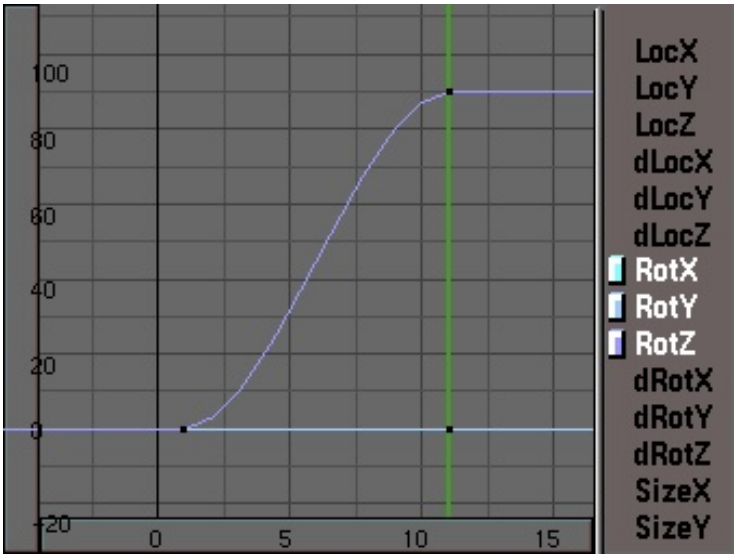
Adding an empty.

Parent the step to the empty. To do this, select the step, hold down Shift and select the empty. Press Ctrl P to parent them and confirm the requester that pops up. To try the result of this little exercise, select the empty and rotate it (R). If all went well the step should now rotate along with it, using the empty as the rotation center.



Rotating the empty rotates the step, too.

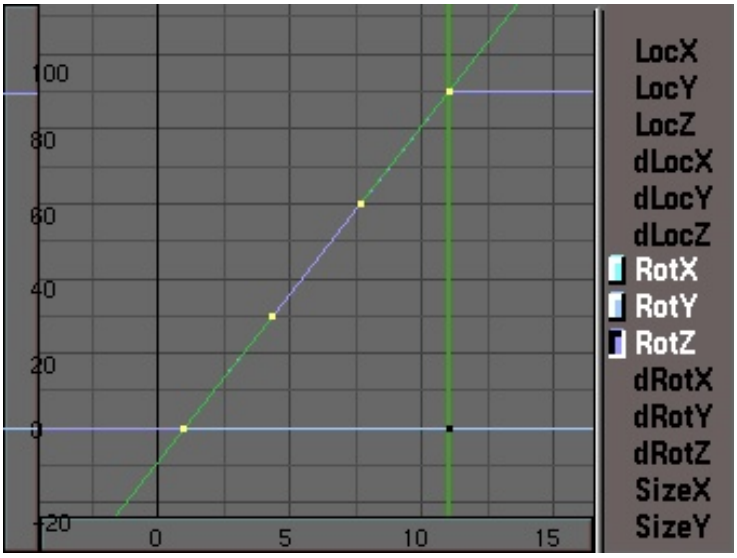
To create a good rotation we will need the IPO window. Split your 3D window and change the new window into an IPO window by pressing Shift[icon F6]. Select the empty and insert a key (I) for the rotation on frame 1. Next, move up 10 frames (press ↑ once). Rotate the empty 90 degrees and insert another key for the rotation. Your IPO window shows a nice rounded curve.



Rotation IPO curve.

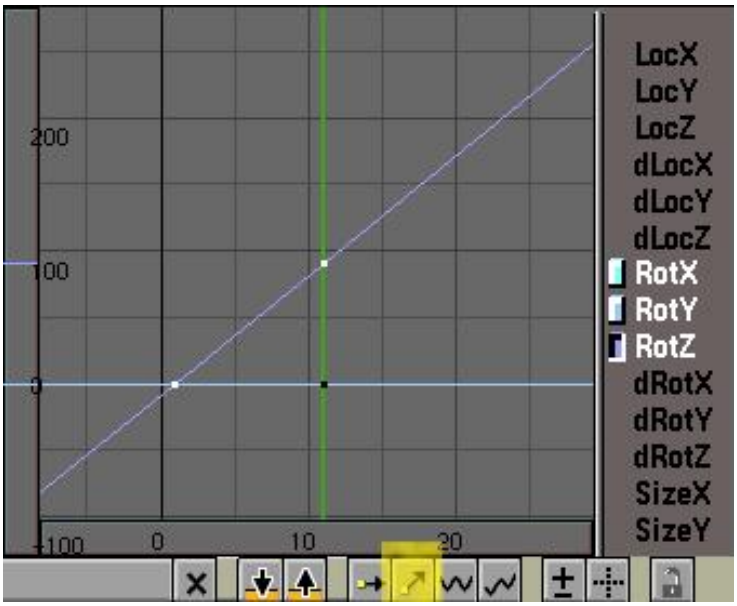
To show your entire IPO curve, press Home.

This is not what I want; the rounding curves indicate an ease-in and ease-out in the animation speed. For a spiral stair however, I want all the steps to be located at equal distances. To fix this, select the RotZ curve and enter edit mode. Select all vertices (A) and press V to make the curve linear. This is starting to look better, but I would prefer the motion to last a bit longer than 10 frames.



Removing ease-in and ease-out.

You have a choice now: you can either move the vertex on the right further away and up, or if you are lazy like me, you can make the curve continue by pressing the extrapolate button: (zoom out first to see the result of this!).



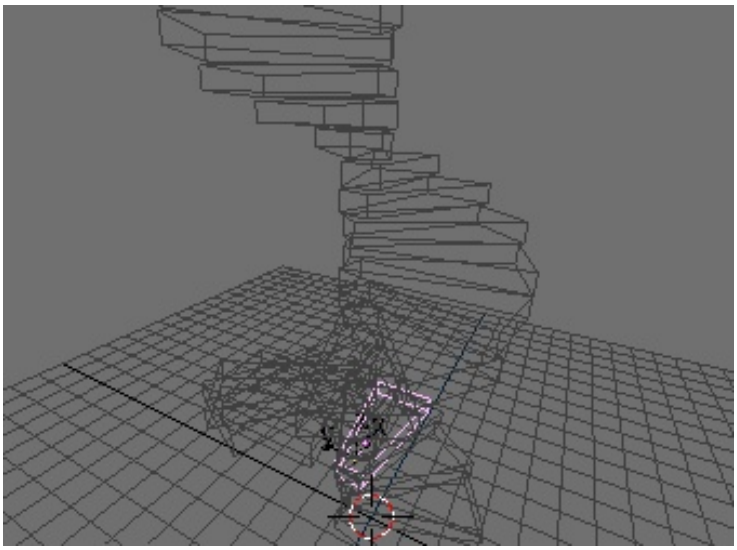
Extrapolating the rotation.

Now to make the step move vertically.. Switch to front view and return to frame 1 of your animation (Shift ←). Select the empty and insert a location key. Go to frame 11, move the empty upward about half the length of the step and insert another key. If you look at the IPO window, you see that the same things happens as with the rotation: Blender gives each motion a smooth ease-in and ease out by default. Get rid of it like you did before, and make the motion continuous.

Dupliframes settings.

And here is the special trick about this tutorial: you can tell Blender to create a copy of an object on each frame of it's animation. Select the step and go to the Animation Buttons window (F7). Select 'DupliFrames'. You immediately see the result in the 3D window:

There are a bit too many duplicates right now. Try playing with the 'DupOff' setting to reduce the number. When you are ready, note the color of the new wireframe: the objects are drawn in grey instead of in black to indicate that they are not 'real'. You can think of the new objects as procedural objects.



Changing the DupOff setting.

Having difficulty with finding the right amount of rotation? Try this: select the empty and turn on the display of the empty's keys by pressing K. Use PgUp and PgDn to select either of the two keys that have shown up. (The unselected key will turn yellow to indicate that it is selected; the selected one remains pink). In this mode, any change that you make to a key is immediately recorded in the IPO curve. Because the objects are still procedural, the entire stair will change along.

To change your objects from procedural objects to real ones, select the original step and press Ctrl A. Confirm the 'Make duplis real?' requester. (Please not that this is not a required action; procedural objects render just like normal objects).

The final result.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=93

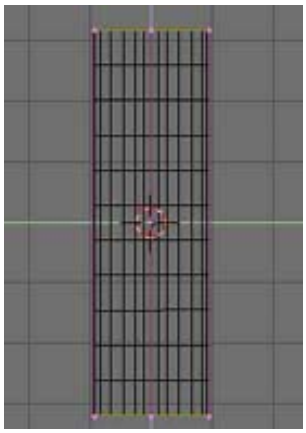
Tutorials
2000 12 11
id93

Candle

Pavel Cernohous

A candle is one of the basic light sources you will need in your scenes, especially when you situate them in the Middle Ages. This tutorial shows how to make the basic shape of a candle and a simple flame. It is supposed that you know Blender interface enought to insert objects and handle them.

I wrote this tutorial several weeks ago for Czech Blender begginers and this is the English version of it. Czech readers please follow this link.

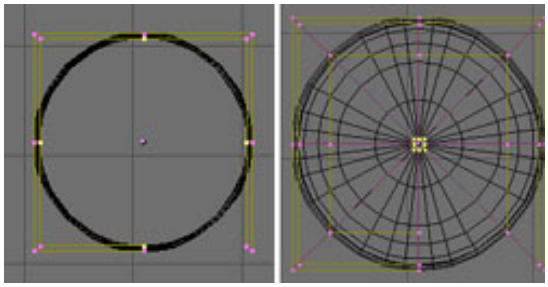


Modelling a candle

Start by adding a tube surface in the top view (NumPad 7) and make it extended in the right view (NumPad 3). You can do this by using box-select (B B) and selecting the top vertices. Next, drag the top vertices upward.

With the top Control Vertices (CV's) still selected, switch to top view (NumPad 7). Extrude by pressing E and scale them down with S. Repeat this three or four times. Please notice that the first extruded circle is only a little bit smaller than the outer circle for the wax will be thin here. Fix this state by pressing Tab twice.

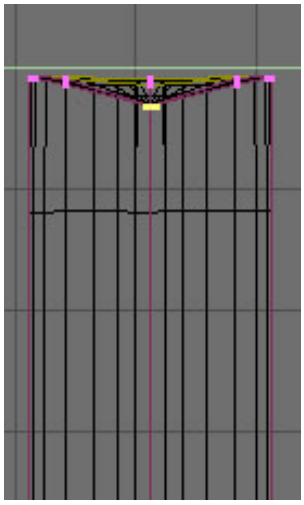
If you are in editmode and want to restore this state later press U, but I am sure you knew this.



Using the magnetic tool

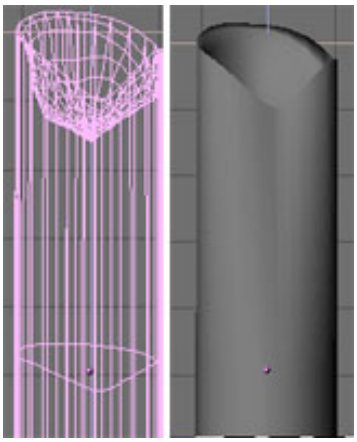
With the border select tool select the middle inner circle and press O to activate the proportional editing tool. This tool works like a magnet and when it is activated actions like moving, scaling and rotations affect not only selected vertices but unselected ones within adjustable range as well.

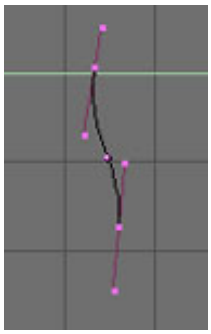
There are two modes for the proportional tool - a sharp falloff and a smooth one. You can switch between them by pressing Shift+O .



Switch to right view now and with the active proportional editing tool move the CV´s down a little. Adjust the magnetic range with + and - keys

Now comes the creative part. By selecting and moving individual CV's with various magnetic ranges, create a crater shape typical for a used candle.





Making a wick

In right view add a bezier curve and shape it as shown. Next add a bezier circle in top view and name it by clicking on OB: field in edit buttons menu (F9)

Now select the bezier curve again and while still in the Edit Buttons window click on BevOb button and enter the name of circle (see above). The cirvle is now extruded along the path. By scaling the circle you can adjust the wick´s proportions.

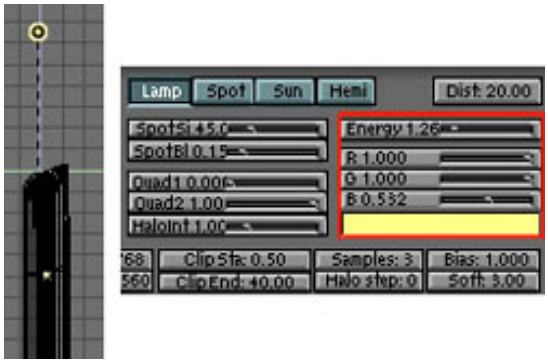
Materials

Let's add some material. For the wick I used a black colored material with Spec and Ref sliders on zero.

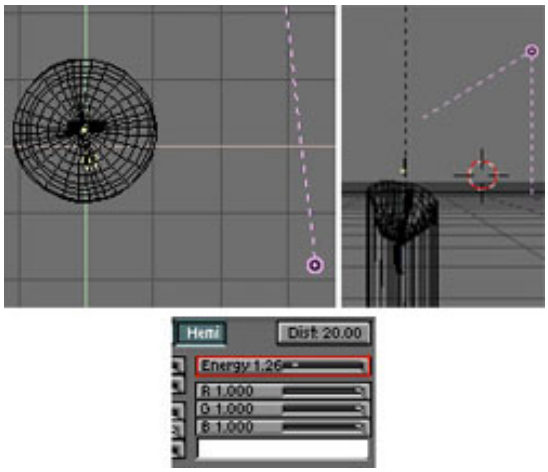
For the wax I used a material with the parameters that you can see on picture. It is a rather dark color we usualy expect for the wax near the flame. This we will make up with lighting

Lights

Add a lamp just above the candle and set it as you see in the picture. This light will simulate the light of the flame as well as giving the impression that the wax is hot around the wick.



The second light is of type 'hemi' and its position and settings you can see on picture again.



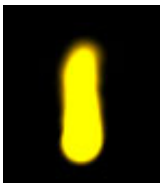
Particles for flame

To add a flame to the wick, set up a particle system. In top view add a mesh circle, scale it down to the wick size and place it where the wick meets the wax. Then in the Anim Buttons window (F7) click on **NEW effect** button then on **Build** button. Change it to Particles.

Here you'll find the settings for the particles. Buttons signed red are relevant. When you press Alt A you should see how the particles flowing from the circle. But when you try to render it, it is still far away from looking like a flame.

Material for particles

To make the flame realistic we will create a suitable material for the particles. First select the circle emitor and add a new material to it. Set the material to a yellow color and turn the Halo button on. Then set halo size. I used 0.60 but it depends on candle size. You may have to try to render it several times to find right value for you.



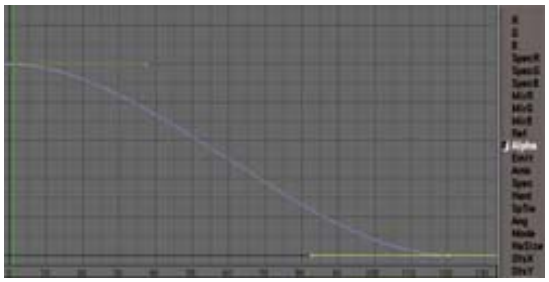
Now the flame should have the right color but its shape is still not not perfect. For a more flame like shape we have to use Ipo editor to change the transparency of the particle during its lifetime. Because the life value for the particles is set to 50 we will make the material almost transparent for this frame. To do this leave the circle emitor selected and press Shift F6 or click on [icon #window.ipo] button. The Ipo editor will appear.

IPO settings

In the Ipo editor you can animate Loc Rot and Size of objects, vertex keys, sequences ipos and for us the most important for now: material Ipos. To switch to material ipos click on the image of the ball at the bottom of the Ipo window. In the left columm you can see all the animateable material parameters. Click on the Alpha parameter for we want to adjust transparency for material. The X axis now defines the frames and the Y axis the value of a parameter.

Did you know...

IPO stands for **I**nter**po**lation.



By pressing Ctrl you can add new Control Vertices into the Ipo editor. Add two CV´s, press Tab to jump into edit mode and select first CV. Press N to activate number buttons and insert values LocX=0; Loc y=100. Handle second CV similary. Values will be Locx=100, Loc y=0 so you´ll get mild gradient for alpha. Jump out of edit mode now and render scene.

When creating IPO's for particles, each particles' lifetime is mapped onto 100 frames in the IPO editor. So even if your flame particle only lives for 50 frames, make sure that you define the alpha curve from frame 1 to frame 100.

Final result!

That´s all. For better results you may play with the material settings and particles settings. You can also add one more particle emitor with slightly different settings from the first flame and add to it smoke material.

This tutorial is translated from czech original on server http://www.grafika.cz/.

Feedback

nozzy 2000 12 12

Very slick!

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=13

Tutorials
2000 07 24
id13

Controlling a Dynamic Object

Bart Veldhuizen

Introduction

After having built a world it is time to add some interaction. In this tutorial, we take a pre-built maze scene and set up a system to control a character in it. The basic principles are rather simple and can be applied to any object in the game engine.

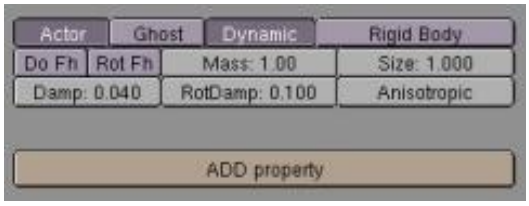
Start by downloading the file below and loading it into Blender.

Download: Maze.zip

Prepare the physics

The first thing to do is to set up the physics for my character; if you don't do this it will not respond to forces such as gravity and friction. (Press P in the sideview window to start the interactive mode and see what I mean - the object just sits there. Hit Esc to stop the game simulation).

Select the sphere and press F8. This will bring up the Realtime buttons window:



The buttons on the left side control the physics of the object. First of all, select 'Actor' to indicate that this is an object that can actually DO something in your game. Two new buttons appear: 'Ghost' and 'Dynamic'. Select 'Dynamic' - this activates Blender's physics engine for this object.

Press P again to start the simulation and watch your object fall to the floor. Because of Blender's built-in physics engine, the collision with the floor is automatically detected and the object is stopped from falling through it.

Sensors

Real-time buttons.

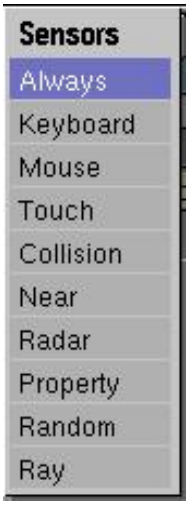
Now it's time to add some interaction. Let's set up a control that will make the character move. Again because Blender has a physics engine, all you have to do to make an object move it apply a force in the appropriate direction when you press a button.

Select the character (right-click on it - it turns pink) to indicate that you want to create logic for this object. At the right side of the Real-time buttons window, you see three columns for sensors, controllers and actuators. These form the heart of Blender's game logic system. Sensors, controllers and actuators are always attached to an object.

Sensors are the ears and eyes of Blender. Create one now by clicking on the 'Add' button next to the character's name ('Character' in the case of the example file). The following line appears.

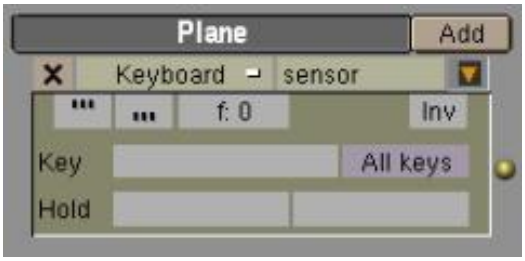
Sensor parameters

Click on the 'X' icon to remove this line from the object's sensor list. Click on the orange arrow to fold up the sensor information. The bottom line contains the information about this sensortype - other sensortypes have different parameters. By default, the sensor type is set to 'Always'. This sensor always returns either true of false. Click on it to select another sensor:



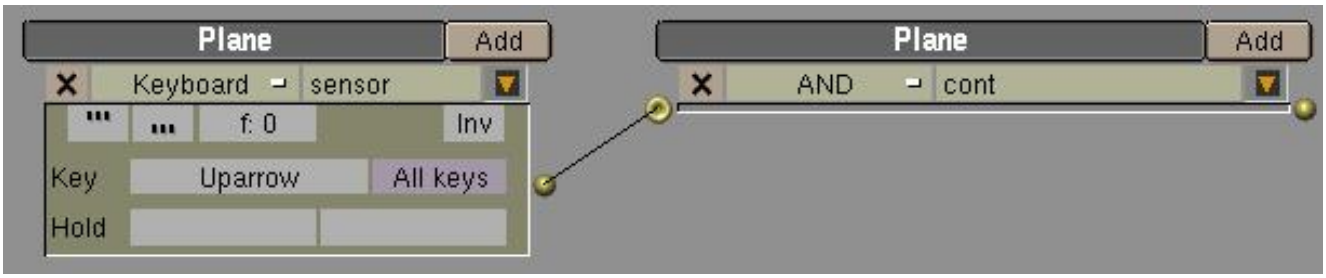
Sensor types.

Select 'Keyboard'. To make this sensor respond to ↑, left-click in the 'Key' field. When it shows 'Press any key', press ↑ to record this key. When you are done, the sensor looks like this:



Recorded keystroke.

Controllers



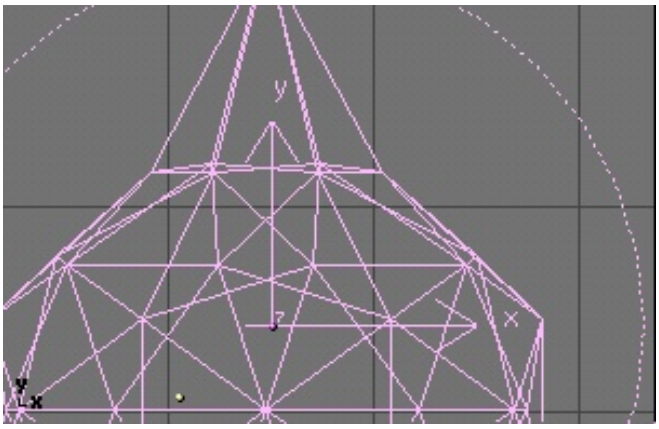
The second column of buttons are the controllers. These are used to combine inputs from different sensors. In our case, there is not much to combine - I only want to pass the key press event to an actuator. Click on 'Add' in the controllers column and leave the controller type set to the default of 'AND'. In Blender, an AND port with only one input just passes the input along.

Next, connect the sensor output to the actuator input by left-clicking on the yellow dot to the right of the sensor and connecting the line that appears to the yellow ring at the left of the controller.

To remove a connection, simply select it by hovering over it with the cursor and pressing x.

Actuators

Now we need to initiate an action. This is done with the third column: the actuators. Add a new actuator now and leave it at the default type: 'Motion'. With this actuator, you can apply different forces to an object, or give it an absolute translation. With all settings you have the choice of which coordinate system you want to use. By default, the Object actuator works in the object's coordinate system. To show these coordinates, select the character and switch to the Edit Buttons window (F9). On the left site, select 'Axis'.



Showing local coordinates.

From this figure, you can see that the forward force should work in the Y direction and that rotations (when pressing left or right) work around the Z-axis. Now enter the value of 10.00 in the Force Y field. This is the second field on the Force line. Connect the actuator to the controller for ↑ and press P to start your game. You should now be able to move your character forward.

What's that? Once you press the ↑ key, your object never stops moving! Thinking about the physics engine, the solution is obvious - just add some dampening! Look at the left side of the realtime buttons window and increase the value of the 'Damp' field.

If you want to have more control over the physical behavior of your objects, add a material to it and select the 'DYN' button in the materials window (F5).

Among other things, you can set the friction and restitution coefficients of a material here. Experiment with them to tweak your environment!

Now go on and create sensors, controllers and actuators for the other arrow keys. For ↓ use a force of -1 in the Y direction. For ← and →, use a torque of 1 and -1, respectively.

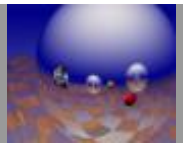
That's all! Now switch to top view and press P to navigate your character through the maze. The completed file is available here.

- To scroll in the real time window, hold down the middle mouse button and drag.
- For a complete overview of Blender's game functionality, please turn to Carsten's gameBlender Documentation

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=100

Tutorials
2000 12 06

EnvMaps



Predrag Kurtovic

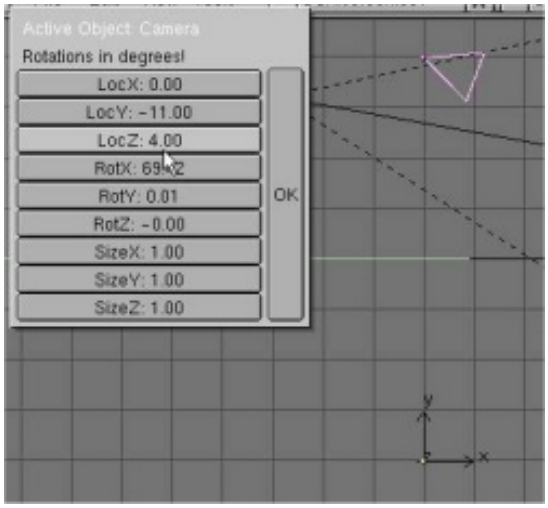
id100

Introduction

Blender uses environment mapping to make reflecting surfaces. Since there are many questions about "how to make reflections" on the Blender website, I decided to write a small tutorial about reflections. I will describe a way to make reflection on a single and/or more objects. So lets start.

Download: intro.blend

Reflection in a plane



The plane is the default object in the Blender scene. Making a Plane reflective is simple. You need: two lights, Plane, Camera and some object above the Plane(text,sphere, ...). Add materials to all object but not to the Plane. Go to the side view NumPad 3, select the Camera and press the N button. You will get x, y and z location for the Camera. The Z location is the most important here - remember it.

Now add an Empty and place it under the Camera. Select the Empty and press N. For X and Y coordinate use the same as for Camera. The Z coordinate value must be same as for the Camera but negative. The Plane should be in the middle (z=0).

Next, add a new material F5 to the Plane. Create a texture for it and select ENVMAP as the texture type. In the "Ob" field enter "Empty" (this is the Object name of the empty - you can see it in the header of the Edit Buttons window - F9). Set the material's mapping type to 'Refl'.

If you now render the scene you will not get a reflection at all; you will get black floor. Why?

This is simple: Blender calculates the reflection by placing a virtual camera at the Empty's location and the Plane blocks the view of the Empty. To solve the mystery I move the Plane to the second layer. Select the Plane and press M, 2. In the Environment Map settings, use the option "Don't render layer" and press second button (this disables rendering of layer 2). Finally enable both layers by pressing Shift 2 in the 3D window. Now when you render you will get a reflective floor (don't forget to press 'Free Data' first - see below).

Download: plane.blend

Environment maps can be of the types 'Static' and 'Anim'. The first means that Blender will calculate the environment map only once per animation; this is useful for scenes in which only the camera position changes - it saves a lot of time.

The second one means that the environment maps are recalculated for each frame of your animation.

If you use 'Static' maps and you have changed something in your scene, select 'Free Data' to force Blender to recalculate the environment map the next time you render.

Reflection in a sphere

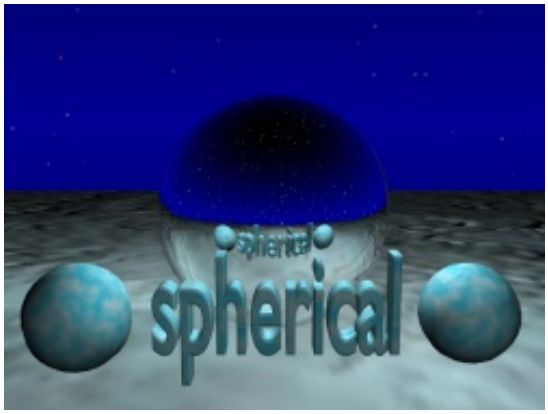
If only one object (in our case Sphere) in the scene has a mirroring surface our job is very easy. Make a scene and add texture to all objects. Now add our main actor - a Reflective Sphere.

Add a mesh sphere. Go to the Edit Buttons window F9 and press "Set Smooth". This way our sphere will have a smooth surface. Apply a new material to the sphere and use the ENVMAP option in Texture options. In the field "Ob" enter the name of your main object, (in our case "Sphere"; again you can see the object's name in the header of the Edit Buttons window - F9). Like above, Blender places a virtual camera at the center of the sphere and uses the resulting image to simulate the reflection.

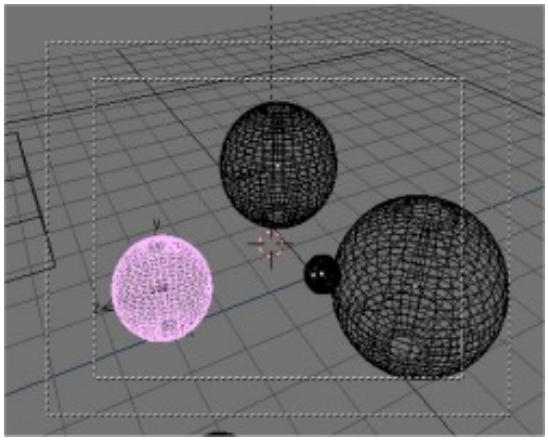
The button "CubeRes" is used to make reflection sharper by changing the resolution. If value of this button is higher then our reflection will be more sharper but need to more time to render.

Switch back to the Material Buttons window F5 and select "Refl" as the mapping type. Only with this settings reflection will be correct. You must also use buttons "Csp"and "Cmir" to add specularity color information to our reflection. If you don't do this our reflection will be too dark.

Download: sphere.blend



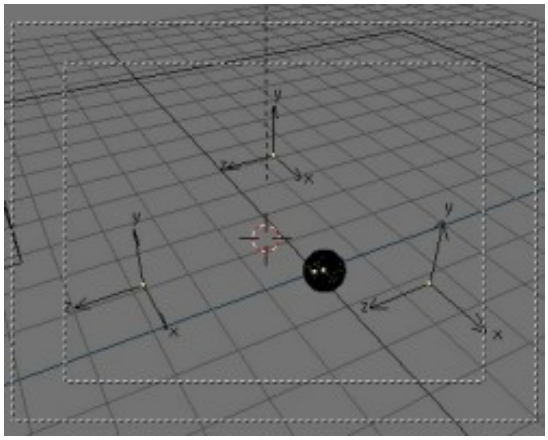
Multiple reflections



Using the tutorial above when two or more reflective objects are on the same scene will not give a correct result. By the law of the physic two reflective surfaces have n-1 reflections. Blender (in the present time) cannot calculate n-1 reflections. If you have three reflective spheres you need to render each EnvMap twice. Why?

When the first sphere create its own environment map it take six snapshots of the scene. The environment maps of the other spheres have not been calculated yet, so the first sphere sees the other spheres without mapping. When the second sphere takes its own six snapshots then our first sphere has a "reflective" but **not true** reflective surface. Only the third sphere has a true reflective surface but with incorrect texures.This is the reason why you must also rerender third sphere's environment map.

Lets start with a clean scene.Press Shift X to erase everything and load the default scene. Add one sphere and **without** moving the cursor add an Empty. Place the cursor somewhere else and repeat this twice. Now you have three spheres on different locations. All spheres have an Empty in their centre. Add Lights and move them together with the Plane, Camera and all other (nonreflective) objects to the fourth layer by selecting all (A) and pressing M 4 and Enter. Add a texture to the Plane.



You can also add some "clouds" by opening the World Buttons window and adding some texture. Be sure to set World to "Real". In other case you will have weird reflections in your objects. On the first three layers you have just spheres. On the fourth layer you have all Emptys (from spheres) and Camera,light and other objects.

Creating true reflection

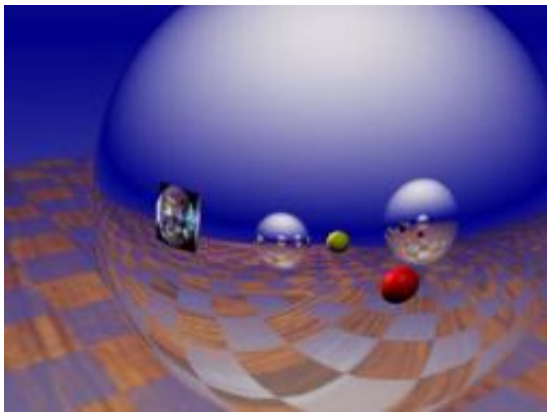
Now add a new material to the first Sphere in the Material Buttons window (F5), and add a texture. Set it to a **"Static"** EnvMap. In the object button write "Empty". Press first button in "Don`t render layer". Repeat these steps for the second and the third Sphere but change the names of Empties (use the header in the Edit Buttons window to find out the correct name, or press 'Name' in the Edit Buttons window to display the name of the Empty in the 3D window). Press the second button in "Don't render layer" for sphere two, and the third button for sphere three.

Press F12 and rendering starts. Now select first sphere, go to Texture menu F6 and **erase** EnvMap with **FreeData** button. Press F12. Blender render EnvMap from first Sphere.Now select second Sphere and repeat procedure-erase EnvMap for second sphere and render it agian F12. Erase EnvMapfrom third sphere.For the last time press F12.Blender calculate third sphere EnvMap. Now all spheres have a true reflective texture.

Animate it!

EnvMaps are nothing more then snapshots of current scene. Because of that you can move the Camera (but only the Camera) on the scene, and all reflections will be correctly rendered. If you move any object, the reflections will not be correct because the EnvMaps are static.

Download: reflect.blend



This little animation show the result of animating a scene with a static EnvMap on an object. Remember you can only move the Camera, not the objects

Download: reflec.mpg

Conclusion/final words/....

At the end of the tutorial, I hope that you have clear picture "how to " make reflection in Blender. Of course you can always ask me if you have trouble using mine tutorial. E-mail is pkurtovic@inet.hr . All parts or tutorial have blend files, and you just need to press F12 to see result. What ever to say then - Happy blendering.

Feedback

josemor	2000 12 07

Is Very Good to know what could count with Blender-NAN to acquire knowledges, over this BIG Softwatre, obviously thank you to people as You. And Bart, etc. etc..... well, to all
It’s very good guide of teaching.
Thank you.

Hadj	2000 12 07

Another winner! good job, Thanks

CyberSteve	2000 12 07

Pretty Cool. I was wondering how to do a reflection.

mikem	2000 12 06

Thanks for this - nicely done.

Toonami_fan	2000 12 06

Great job! It explains perfectly=0)

<MATT>

D@n-Stryder	2000 12 06

Nice tuturial. I was just going to look for a tutorial on EnvMaps

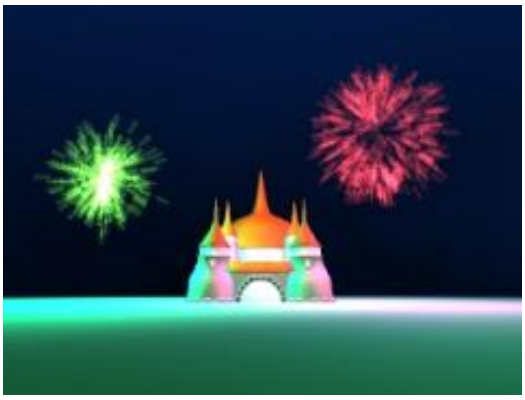
www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=111

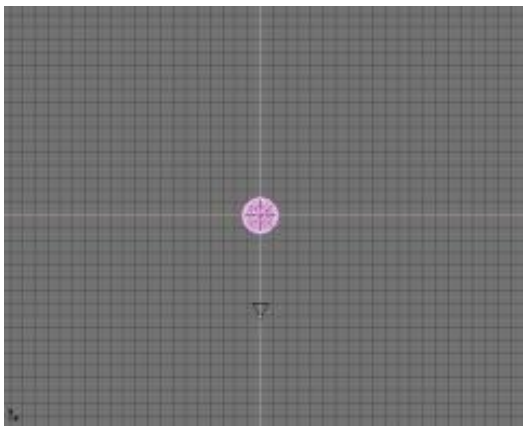
Tutorials
2000 12 21
WP
id111

Fireworks

Introduction

Well its almost that time of year again when fireworks will brighten up our skies, so i thought it might be a nice topic for a tutorial. A couple of days and empty soda bottles later you can try it out for yourselves. Have fun reading it!





Getting started

We will start off by deleting the default plane, simply done by selecting it and hitting the X, now add a UV sphere by hitting the spacebar and selecting Mesh from the menu. In the following menu choose UVsphere; this sphere will be our particle emitter.

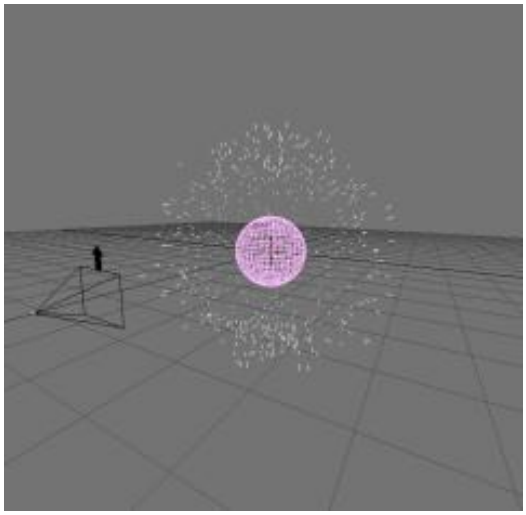
To add particles to this object we'll hit F7 to go into the effects menu. Press the New effect button and choose the "particle" option in the pull down menu on the right. Voila! A lot of new buttons to play with. Don't be scared - we will only use a few of them in this tutorial!

playing with particles

First we will set the total number of particles to 500, this is simply done by holding Shift and pressing on the "Tot:" button which is located in the top left corner of this menu (it is set to 1000 by default). Next we will set the start and finish frame on which the particles should be emitted from the sphere.

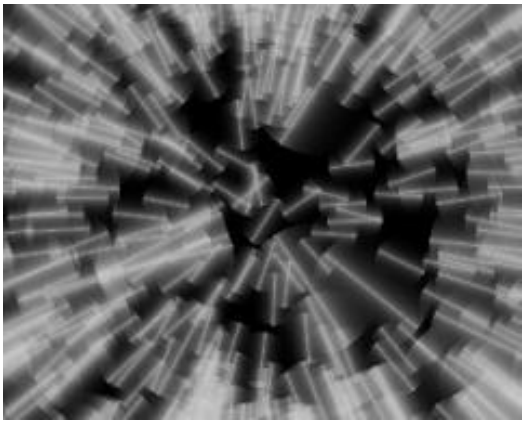
Set the start frame to 25 and the end frame also to 25. This is done because there is only 1 sudden explosion of particles; if you are creating a fountain you can set the end frame a number of frames later to create a more spray like effect. Both the start and end buttons are located to the right of the 'Tot:' button. Now look 2 buttonrows down for a button called Bspline and press it. The Bspline will make the motion of the particle smoother. Also press the Vect. button to the right of it. This will change the particles into lines instead of the normal points, which looks better for the look of firework were going for.

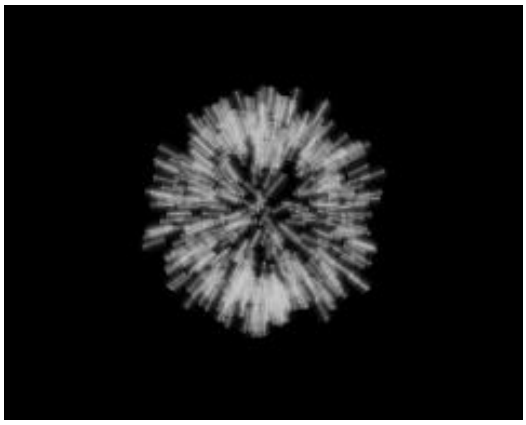
Now we will give the particles different forces to make it act a bit more like firework. Look in the purple colored row of buttons, the first one is called Norm: this stands for normals, and is for the initial speed at which the particles will be emitted. Set it to 0.300. The Rand: button is located 2 buttons to the right , which is for the randomness in the particles. We don't want the particles to be emmitted in perfect straight lines because that doesn't look natural; to add some chaos to the particles set this button to 0.200. The last button in this purple row is the Damp: button, this button dampens (you might have guessed this:) the initial speed of the particles, so instead of always going the same speed, the further away they get from the sphere the more they will lose speed until they have none left. Set the button to 0.600.



The last thing we need to do for the movement of the particles is to add a little force to them to simulate gravity doing its job, look in the lower left corner of the fx menu, 3 buttons called X, Y and Z are located there, Set the Z to negative fifty, or -0.050 in numbers. If you hit Alt A now you can see the animation, as you might see the particles don't seem to appear from every vertex on the sphere, to fix this little problem go into edit mode by pressing Tab and select every vertex by hitting the A key. Next up we press F9 to get into the edit menu, now look for a button called Hash on the bottom row, once found we press it :) We go out of edit mode by pressing Tab once again. Hit Alt again and voila a burst of nicely distributed particles!

If you press F12 now to render an image you will see a big mess off particles, this doesnt look very nice now does it? this is mainly because the camera is to close to the action and because we havent assigned a material to the particles yet.

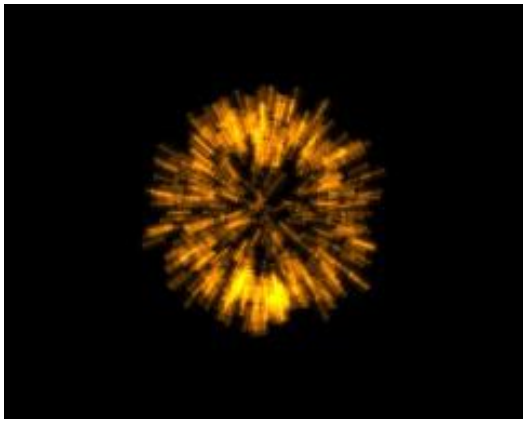




To fix the first problem make sure you're still in topview (NumPad 7), select the camera with RMB and grab it with G while holding down Ctrl. Now we can move the camera 1 grid unit at a time, move it 15 units down and click RMB to place it there. Render another picture, much better if I say so myself :)

Now on to fix the second problem: the material. Make sure you have selected the sphere RMB and hit F5 to head to the material menu, Once arrived we will add a new material, this is done by clicking on the button with a '-' sign on it.

First we'll select a color, i chose orange, but you can choose what ever color you want as long as their bright. Once we've done this look to the right of the RGB sliders to the purple vertical row of buttons, now push the one that says Halo, this is a special material in blender meant for particles. To add the final touches to the material look to the left for the Alpha, Hard and Add sliders. Set Alpha to 0.650, Alpha is for the transparency of the halo, Hard to 127, this makes the halo look sharper and Add to 1.000, add will make it look like the halo is emitting light itself, which is what we want in our firework.



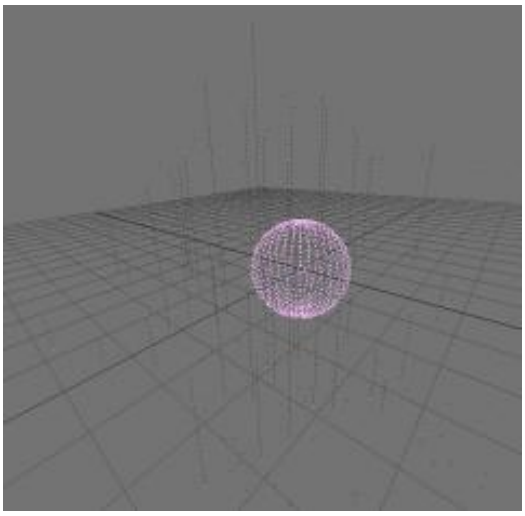
When we now render an image F12 we can see how much improvement we have made over the previous rendering, As a nice touch to the material we will animate the Alpha value to let the halo's fade out as time passes. Make sure you're at frame 25, you can skip through frames with the ← and →. With the mouse in the MaterialButtons hit the I to insert a keyframe, a menu will appear with a number of choices, choose Alpha. Next up we will go to the last frame that the particles are visible on, in this case thats frame 74. Once arrived on frame slide the Alpha slider all the way down to 0.000 and insert another alpha keyframe. Were almost there!

To add the final touch to our firework we are gonna be using a second particles system with lamps dupliverted onto them. It sounds like a mouthfull and maybe even complicated but it isnt (trust me on this). We need this final step if we want to add this firework into a scene, real firework gives of light and ours doesnt yet, and thats what were gonna fix right now.

Start off by duplicating the original sphere, since we want our lamps to have the same motion. Next up we will reduce the number of particles in the second sphere, cause if we are going to be using dupliverts for this trick every particle will be replaced by a lamp, every particle you may ask? yes every particle :) Now if we didn't reduce the amount of dupliverted lamps the increase in rendertime would be quite dramatic as would the drop in framerate in youre 3d window. Of course if you want to you can knock yourself out with the amount of dupliverts or particles you can add. Just remember that if you're using a lot of dupliverted lamps to keep the original lamp at a very low energy level because every dupliverted lamp will have the same amount of energy.

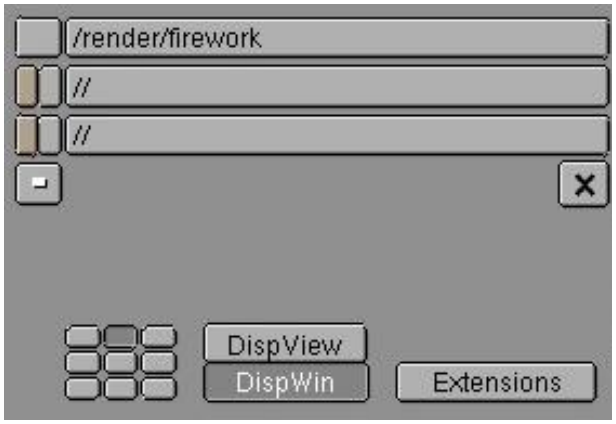
With the new sphere selected, go to the fx menu F9 and reduce the number of particles from 500 to 50. Add a lamp by hitting the spacebar and selecting lamp. Now select the lamp with RMB and while holding Shift select the sphere (also with RMB). Were now going to parent the lamp to the sphere by hitting Ctrl P. Go back to the fx menu F9 and look on the left side, look for a grey button called DupliVerts and push it.

If you now look in the 3d window you will see that every particle emitted by the second sphere has a line beneath it, this is to indicate that the dupliverts are working. Select the lamp and go to the Lamp menu F4 we will now play around with the lamp settings, the first thing we will do is lower the energy from the default 1.000 to 0.150. this will be more then enough (remember 50 lamps times 0.150 energy still creates a lot of light :)



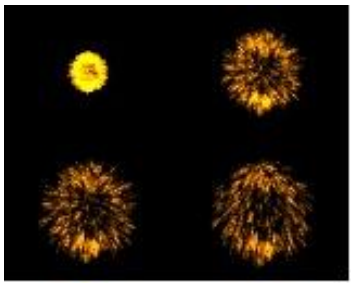
Furthermore we will also add colour to the lamp by simply sliding the RGB sliders around, once again i chose a orange color since my firework material also has the same color. As a final touch to the lamps we will also animate the energy so that when they firework fades out the lamps will do the same, go to frame 25 and hold youre mouse over the lamp menu and press the I to insert an energy keyframe, now go to frame 74 set the energy of the lamp to 0.000 and insert a second energy keyframe.

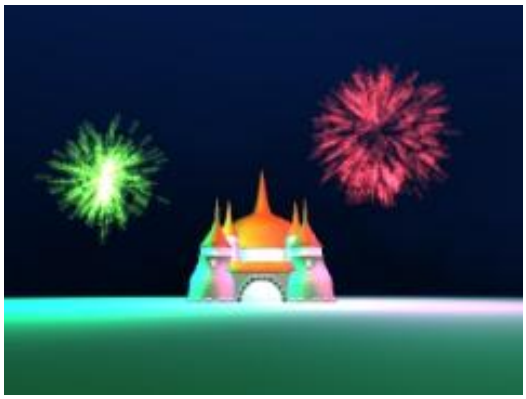
Your display of fireworks is now ready to be put in a scene! or you can render an animation to enjoy what youve created right away. To finish this up I'll take you through this as well (yes i know, i'm a nice guy ;). Press F10 to go into the render menu. First we will select an output format, Aviraw is a good choice because most mpg encoders can handle it. Furthermore set the end frame to 90 instead of the default 250, otherwise we would be rendering a lot of black frames which we dont need. Beneath the big Render button is a button located called OSA , which is short for Oversampling, osa filters and anti aliases the final image which basicly means you're pictures will look nicer :)



Now before we hit the Big Anim button to render out the animation we need to give a name and path to the animation. This is done on the left side of the menu; there's a button located which says **/render/** (not the big pink one). Click on it and fill in a name you want for the animation; I named it firework (yes I know, quite corny :) now hit Enter to confirm the name, and YES we can finally press that big shiny anim button. Come on just press it, you know you want to! Allright while the animation is rendering get yourself a beverage and congratulate yourself on completing this tutorial.

Well give yourself a big cheer cause you've done great getting this far! To really wrap it up you might want to make a scene to place you're fireworks in, after all we're using lamps and lamps need objects to bounce their energy off. I've gone ahead and created a little scene for you.





In the first layer is the castle with some night lighting on it, in the 2 second layer are my fireworks, by simply duplicating the original piece of firework and giving them different colors and start and endframes I've created a variety of them. You can acces the layers by simply hitting the 1 and 2 keys, to have them both on at the same time select both layers while holding the Shift key. You can download the scene right here or watch the mpg of it below. Hopefully you've learned some new features or just had a good time reading the tutorial.

Download: fireworks.zip

Download: firework.mpeg

Feedback

Nick 2000 12 22

impressive. thank god for blender.

Pol 2000 12 21

Sweet !

nozzy 2000 12 21

Nice tut..

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=133
News

Flock Simulation Plug-in

2001 01 22

Tatsuya Nakamura

id133

What is Flocking?

Flocking is a very famous teqnique for simulating group behavior. I'd like to quote the good description about it from "Game Programming Gems" (Charles River Media):

Flocking (sometimes called *swarming* or *herding*) is a technique first put forth by Craig Reynolds in a 1987 paper he did for SIGGRAPH, "Flock, Herd, and Schools: A Distributed Behavioral Model." In that paper, Reynolds proposed a series of three simple rules, which, when taken together, gave group of automous agents (also called *boids*) a realistic form of group behavior simular to flocks of boids, schools of fish, or swarms of bees. These rules, which Reynolds refer to as *steering behaviors,* are:

- **Separation.** Steer to avoid crowding local flockmates.
- **Alignment.** Steer toward the average heading of local flockmates.
- **Cohesion.** Steer move to toward the average position of local flockmates.

In later implementation and papers, Reynolds added what has sometimes been referred to as the "fourth rule" of flocking:

- **Avoidance.** Steer to avoid running into local obstacles or enemies.

Note that the steering behavior say nothing about state information or about a given agent maintaining knowledge of the flock, its environment, where it:s headed, or the like. Flocking is a *stateless* algorithm in that no information is maintained from update to update; each boid reevaluates its environment at every update cycle. (*written by Steven Woodcock*)

You can easily find more detailed descriptions from "father of flocking" Mr. Craig Reynolds' website http://www.red3d.com/cwr, or other useful resources on the web.

What is this Plug-in?

This is the Python Plug-in for Blender to see the results of the flock simuation on 3D window. The original source code of this plug-in is done by Mr. Christopher Kline and his imprementation is one of the excellent references on this subjects on the Web. His *"C++ Boids"* is platform-independent and soure code is available public from his website.
You can check out his site to see details of about his Boids. His demonstration program based on Open Inventor is also available on the site. I wrote simple Python-C wrapper (not C++, for quick and simple imprementation) to use the library on Blender.

How to use this?

You can download plug-in and sample blend file from here.
At this time, I made executable for this plug-in for Windows and IRIX6.5(SGI). These include original source code, my Python wrapper code, and sample Blend file.
Linux executable is also available. Please see the later part of this document.

Download: flockWindows.zip

The IRIX version is available here.

Download: flockIRIX.tar.gz

To run the sample file, you extract the archives (please read README!) and put the flock.dll (Windows) or flockmodule.so (IRIX) in the directory as you like. (Now 'C:/Blender/flock')
You run Blender and open 'flockSample.blend'. You can see script window on the right of the Blender screen. You have to set the sys. path to your directory that you put the flock.dll or flockmodule.so.

```
# flock.py

import sys
# set the path to your directory
sys.path = ['C:/Blender/flock/', '.']
import flock
import Blender

num = 20
fr = Blender.Get('curframe')


if (fr==1):
    flock.makeBoids(num)
    s = Blender.Object.Get('Sphere')
    flock.setSphere(s, 4.24)
    b = Blender.Object.Get('Box')
    flock.setBox(b, 10.00, 10.00, 14.16)
    for i in range(1, num+1):
        name = "Plane.%03d" % i
        ob = Blender.Object.Get(name)
        flock.setBoid(ob,i-1,4,25,1.0,1.0,1.0)

else:
    for i in range(1, num+1):
        name = "Plane.%03d" % i
        ob = Blender.Object.Get(name)
        flock.updateBoid(ob, i-1, fr)
```

Next, You have to attach this script to 'Frame Changed' as left. The script will be called every frame. Push Alt+a on 3D Window and you can see the boids movement.

How to use it in your scene?

This module "flock" has following methods:

- **makeBoids**(n): n=the number of boids (integer)

Initialize data for all Boids. You have to call this at fist.

- **setBox**(o, x,y,z): o=object reference(Blender.Object) (x,y,z)=the position of a vertex (double)

Set a box-shape obstacle. You have to choose an arbitrary vertex position from the six vertexes to decide the size of the box. (In Blender, push [N] to get the vertex position information after select it in edit mode.)
The all edges in the box must be parallel to one of three axis in global coordinate. This means that its diagonal line is from (x0 +x, y0+y, z0+z) to (x0-x, y0-y, z0-z) when the center of the box is at (x0, y0, z0).

- **setSphere**(o, r): o=object reference (Blender.Object) r=radius (double)

Set a sphere-shape obstacle.

- **setBoids**(o, i, v0, v1, dx, dy, dz): o=object reference(Blender.Object), i=boid id (integer), v0=initial velocity(double), v1=max velocity(double), (dx,dy,dz)=dimension(double)

Set Boids' initial values.

- **updateBoids**(o, i, f): o=object reference(Blender.Object), i=boid id(integer), f=frame (integer)

Update Boids positions and velocities per frame.

For detail, please take a look at the sample script in the sample blend file. Basically, you makeBoids and setBoid at the first frame and run the simulation from the next frame as long as you want to do.
When you use your own model as boids, you have to set your model's local Y-axis as forward-backword and local Z-axis as top-bottom.

Limitations and known problems

- This module for Blender 2.04 and earlier and **does not work for Blender 2.1 beta**.
- **No saving functionality**. If you want to save, please write positions and velocities of boids each frame in a text file using Python.
- The boids has no state information and their positions will not resume when animation frame get back to the first frame. For instance, when start frame is 1 and end frame is 250, boids will be animated eternally in the loop from 1 to 250 frame until you stop.
- In original source code includes more detailed parameters to control flocks. In this version, all you can define is initial positions, initial velocity, max velocity, and dimensions of boids.
- Play back repeatedly causes performance problem in Python script execution.

How to make executables on your OS?

I think it is easy to build the binaries for other UNIX-like platform with GCC. Please see Makefile include source codes in IRIX archives. In fact, Hiroshi (thanks!) successfully compiles the source on Linux. Please download from here.

Download: flock.so.gz

Future work and Acknowledgement

As I mentioned before, some parameters shoud be controlled thru Python API. Of cource, you can modify and re-compile the source code to expand its functionarity.
This version is for flying objects. (Literary, Boids means Bird-oid.) To apply this algorythm to horses running on the grounds, you have to add kind a ground path though functionality to the souce code.
There are much references on the web and they describes bettern than me;).
I'd like to thank you for **Mr. Christopher Kline who gave me the permission to publish this plug-in.**
Hope you enjoy the flocking on Blender!

Feedback

wedgles99 — 2001 01 23

More! More! More!

Qbertino — 2001 01 22

Yeah, dig it! Some solid 3D pro-level stuff here. Sounds like Character Studio 3. Can we have direction and pace-dependent animation for idividual Boids next week please? :-)))

James15185 — 2001 01 22

Cool... I like plugins. I hope to see more of them.

GammaRayQ21 — 2001 01 22

Knowledge is what keeps me alive

heappies — 2001 01 22

Great!...you have enhanced my knowledge

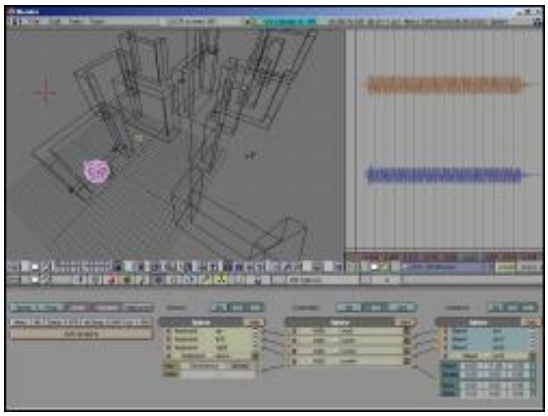
www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=14

Tutorials
2000 07 25
id14

The Blender Game Engine

Bart Veldhuizen

Introduction

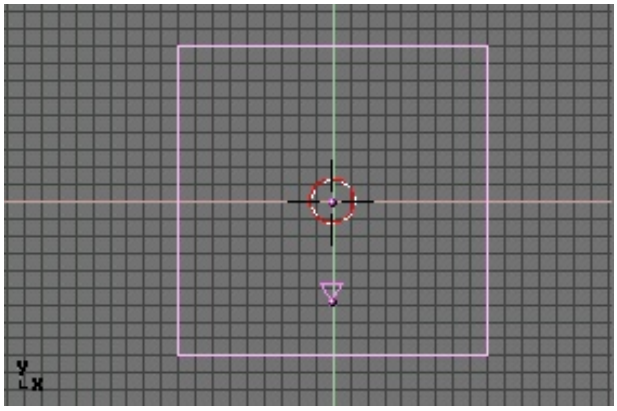


This tutorial explains the basic concepts of Blender 2.1 It assumes basic knowledge of how to use Blender - these have not changed much since Blender 1.8 and the old tutorials are still applicable (although sometimes a button may have been moved to another location).

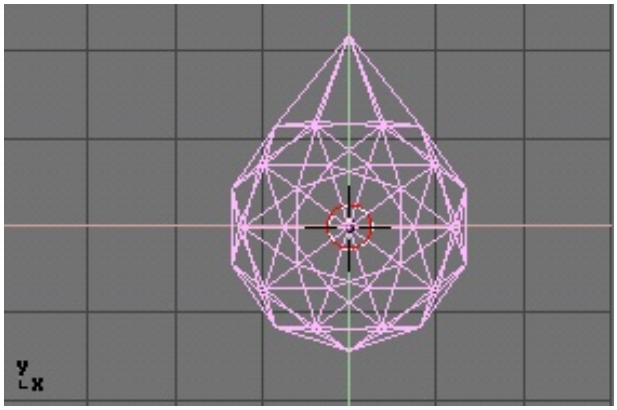
If you are unfamiliar with Blender, please do the following tutorials first:

- The Blender User Interface
- Navigating in 3D Space
- The Blender Windows
- Building a Castle

Let's start by adding a cube. Remove the default plane and add a cube. Start scale mode and make it large enough to fill the entire screen. We'll use the cube as the scenes floors and walls.



Let's add a game character now. In top view, add an icosphere with a subdivision of 2. (The higher the subdivision, the more polygons your character will get and the slower your game can become!). In editmode, drag out a 'nose'. This will indicate our direction of movement later on. Moving around with a perfect sphere is rather confusing.



Press P to start the real-time simulation. It just sits there - nothing happens! No wonder, since we have not defined any physics or events yet. Let's do that now.

Dynamics

Stop the simulation with Esc. Select your character and go to the Real-time buttons (press Shift F8). On the far left you'll see a button labeled 'Actor'.

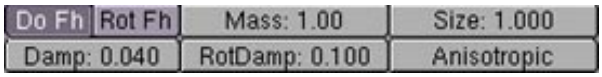
An actor is not necessarily the main character in a game. An example of an actor can be an enemy that follows you around the game.

Select Actor for your character. If you would restart your simulation at this point, still nothing would happen. Let's turn on the physics simulator for this object now: with your character still selected, press 'Dynamic'. ('Dynamic' appears after 'Actor' has been selected).

Rotate your view a bit (the top view will not show this effect properly) and press P. Your character falls to the floor.

Forces and Materials

Now let's work on the physics a bit more. First of all there is a global parameter for gravity in the World Buttons window. Add new world settings; a new slider called 'Grav' is now availabe. This is a constant for your world; all the Dynamic objects are affected by it. The standard value is 9.8, which results in a natural gravity-like acceleration of falling objects.



Now return to the Real-time buttons. For each object, you can define the a number of parameters.

1) Mass.

When applying a force, the mass determines the acceleration that an object gets. Higher mass results in a smaller acceleration.

When you have both large and small objects in your game that interact with each other, be sure to match their relative masses - this will add to the realism of your simulation.

2) Damp.

Motion dampening. This is a friction force that always works on your object - both when it touches a floor and when it is free. A good way to describe it is 'air friction'.

When an object has a Damp value, it will always slow down when there are no other forces working on it

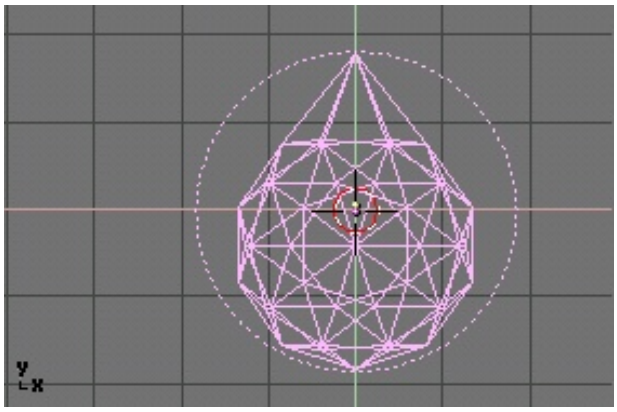
3) RotDamp.

Rotational dampening. Like the motion dampening, this affects an object's rotational speed.

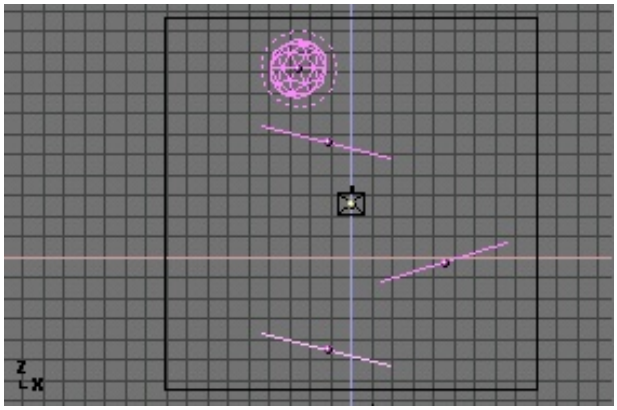
4) Size.

Blender's current physics engine uses spheres to represent your dynamic objects. This parameter represents the size of the sphere. The collision sphere is drawn as a dotted circle in the 3D window. In most cases, you should try to enclose your object in the collision sphere. If you don't, parts of your object will be able to stick through walls and other objects.

Set the size value now for your character. The collision sphere will center around the icosphere's object center. To shift the object relative to the collision sphere enter editmode, select all vertices and move them around.



Let's play with the physics simulator a bit more before we move on. Add a plane while in topview and switch to front view. Create some duplicates and rotate and move them around a bit until you have a situation like this:



Press P and watch your character fall down the different planes. Something still is not right here: whenever it hits a plane, it immediately sticks to it and starts to slide down. To fix this, we'll have to tweak the dynamic material properties of the planes.

Materials

Besides color information, each material now also has dynamic properties; using these you can define different forces that define the interaction between the objects that carry this material and dynamic objects.

Add a new material to one of the planes now and select the 'DYN' button.

As the first test, let's play a bit with the 'Restitut' value. This value determines how much of the object's kinetic energy is returned to the object after it has a collision. In other words: this value defines the 'bounciness' of your material. Set it to one now. Select the plane directly underneath your character and clear its rotation with Alt R to level it. Press P to play your simulation.

You would expect it to keep bouncing, wouldn't you? Wrong - remember the 'Damp' setting? The dampening slows down your sphere a little all the time. That is why it bounces less and less high. Select your character and set 'Damp' to zero. Press P again and watch it bounce ad infinitum.

Take a look at this example: the planes have been placed to keep the ball bouncing around for quite a while:

Download: bouncing.zip

Reset the 'Damp' value to 0.04 now. The next thing we are going to try is called 'Fh'. This is a term that has been borrowed from physics where it means a force in the vertical direction. It works the same in Blender, but here we also have to define a region of influence. This region of influence is controlled by Fh Dist. This value creates a 'cushion' around your object in which the Fh force will be active.

Try the following: set Fh Dist to and Fh Force to 0.4. Now select your character and go to the Real-timeButtons again. You have to click 'Do Fh' to enable Fh. Now play your game (clear the rotation of the top plane again before playing). Your character now hovers above the plane like it is floating on a magnetic field.

The influence of the Fh force is linear with the distance to the object: it varies linearly from zero to Fh in the interval of Fh Dist.

The Fh Dist value is also used for collision detection. As soon as another object enters the 'cushion', a collision is registered. What's more, the incoming object gets the color of the object that it collides with. This allows for visual debugging during playback.

That was it for the basics of the game engine. The next step is to learn how to control your character. This is described in Controlling a Dynamic Object.

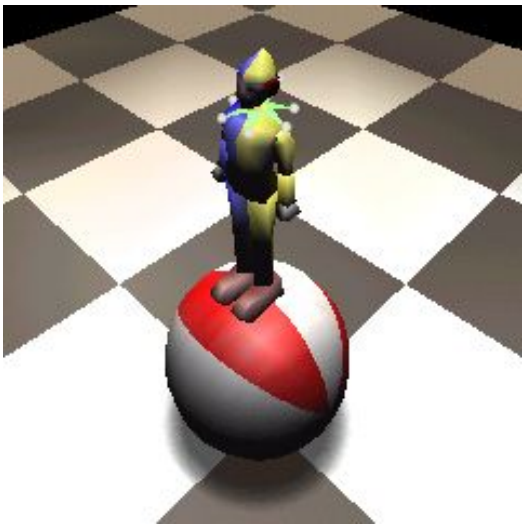
www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=177

Tutorials
2001 05 16
id177

Game Textures

Yamasaki Satoshi

Introduction



gameBlender has many effects for the realtime rendering engine. We can make cool visuals to use them more effectively.

In this tutorial we will use them to make a nice looking clown.

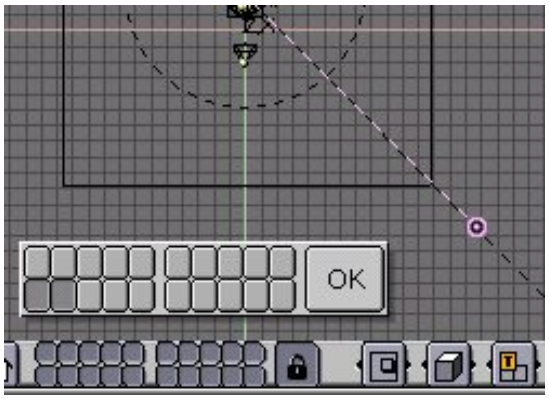
Begining

First you need to download the file below (clownsmp.zip).

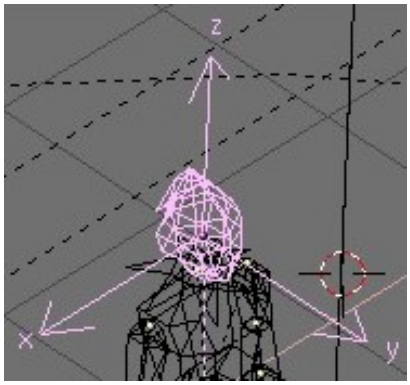
"clownsmp.zip" includes "clownsmp.blend" and "texture" directory (you must unzip "clownsmp.zip". Now load "clownsmp.blend".

The left window is the 3d window in wire-mode (Z). The right window is the Image window for displaying textures.

Download: clownsmp.zip

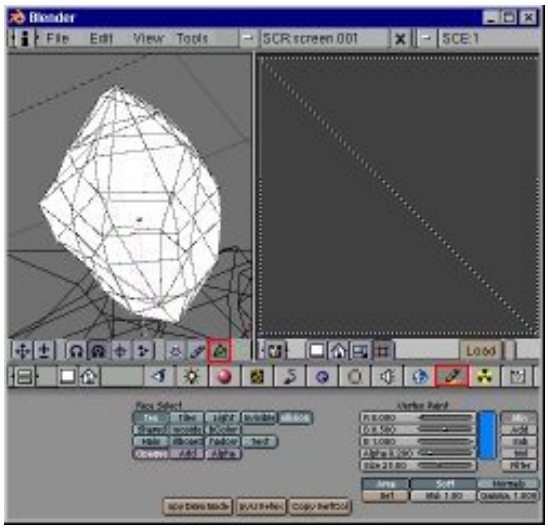


I put three lamps in this sample. Select a lamp and press M key. You can see the layers in which the lamp is present. You have to put lamps in same layer as the object that you want to light. You can select multiple layers for a lamp (click the layer buttons while holding Shift)

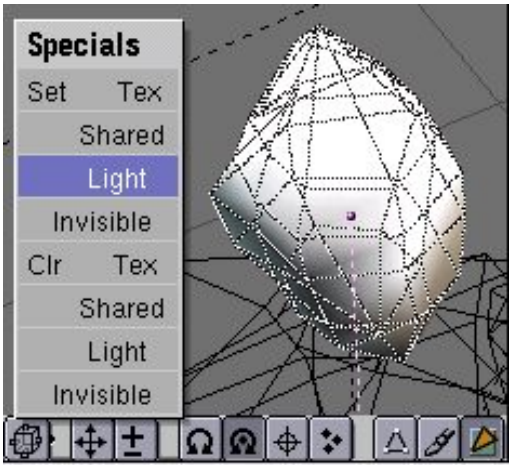


Now select the head. If you make objects in gameBlender, the direction of the object is important. Press the **Axis** button in EditButtons and look at the 3D window (it should look like the screenshot when you have the button pressed). The front is indicated by the Y-axis of the object. It is used for "Track to", UV-coordinates, etc...

Selecting Faces



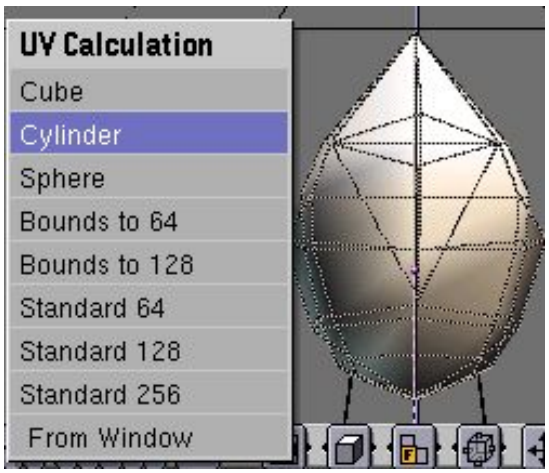
Enter the Face Select Mode (F key) and change the ButtonsWindow to "Paint Buttons" (click icon). If not all of the faces are selected, Select All with A.



Press W key with the mouse in the 3D Window to open the Specials Menu. Now we can apply a command to many faces at a time.

Select the first **Light**. This command allows faces to receive realtime lighting from the lamps we set up (**Note:** when you select the second **Light** the switch is set to off).

You can see that the object is drawn in shaded mode. Also look at the Buttons Window and make sure the "**Light**" button is pushed.

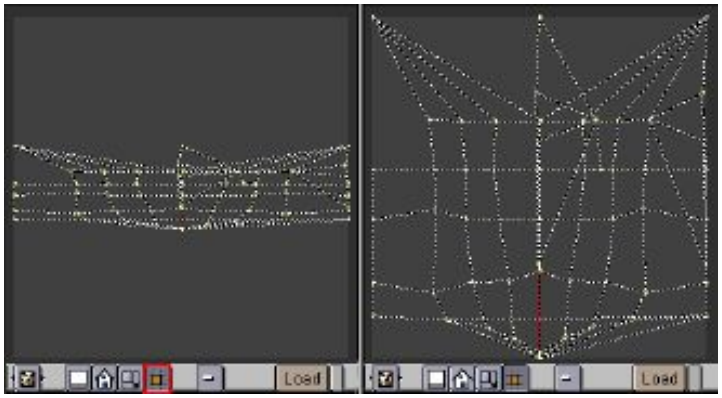


UV-Mapping(1):UV-Calculation

Now it's getting about time to attach some textures to the faces.

To do this we need to set the view to rear view first. Click icon in the 3D Window Header while holding the Shift key.

Press the U key to bring up the "UV Calculation Menu" in the 3D Window.

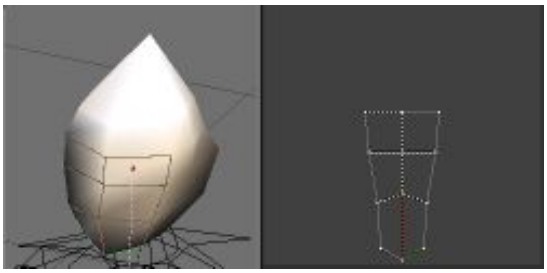


Select **Cylinder**. Blender will show the UV Mapping in the Image Window. This UV-map is too small, so we'll have to resize it:

- Click the "Clip Image Icon" (marked with red in this figure).
- Select all the vertices with A and Size S them in the Image window so they fit the area you want.

If you want to paint a texture image, make a screenshot of the ImageWindow (Ctrl F3 with the mouse over the ImageWindow). You can use this screenshot as a template for the texture.

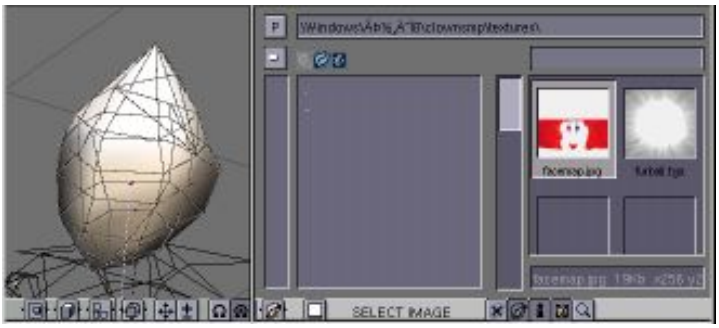
UV maps can be pretty complex. You can paint textures more easily if you only select the faces that will need a certain texture before making a screenshot.



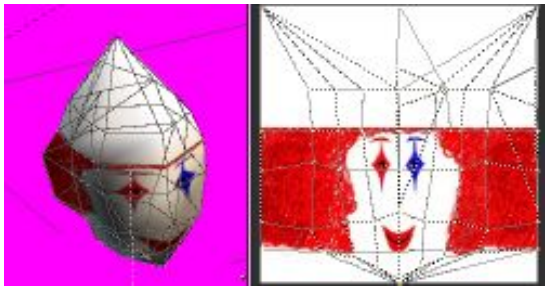
UV-Mapping(2): Attaching a Texture

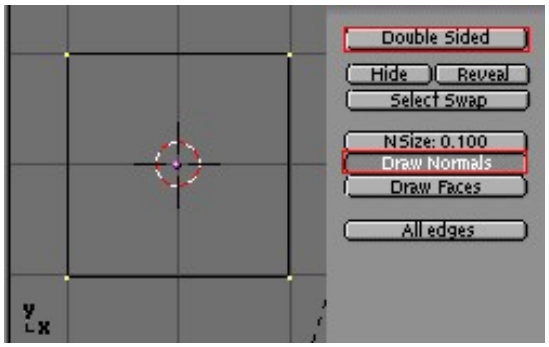
We'll use the sample file "facemap.jpg" from the zipfile you downloaded earlier in this tutorial.

Press "**Load**" button in the Image Window, and select "facemap.jpg" from the texture directory.



Change to "Potato-mode" (Alt Z) in the 3D Window. You'll see the texture has been applied to the object. When you're done press F to exit Face Select Mode.





Halo effect

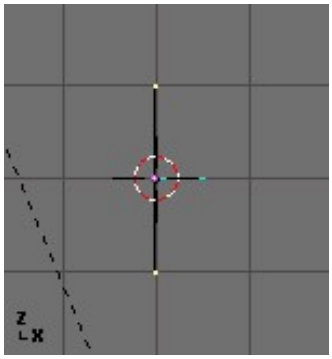
The "Halo" effect makes each face turn to the camera. It is often used for flares, lightning, and in this case something like a ball. Let's make the clown's nose and the balls on his collar and hat.

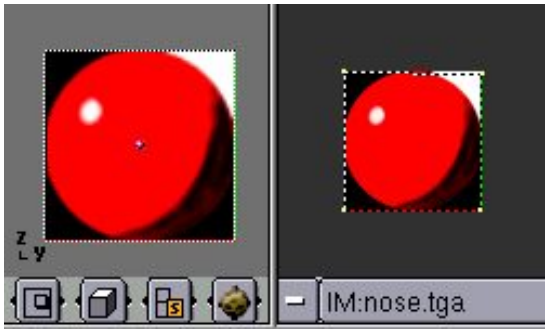
Change to top view (NumPad 7), and add a Plane Mesh (Shift A >> MESH >> Plane) somewhere.

Make sure the **Double sided** button is *not* checked, and **Draw Normals** is checked in the EditButtons F9.

Next, Change Front view(NumPad 1), and Rotate (R) all vertices to 90 degrees (hold Ctrl).

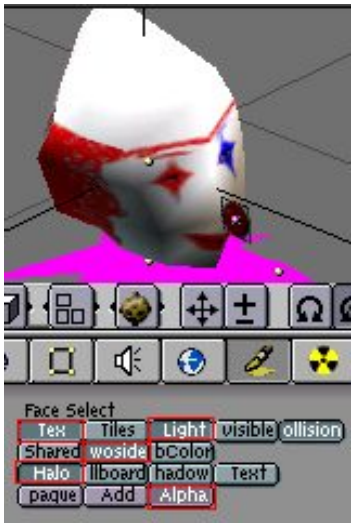
Note: Halos will only show faces that face the positive X-Axis.



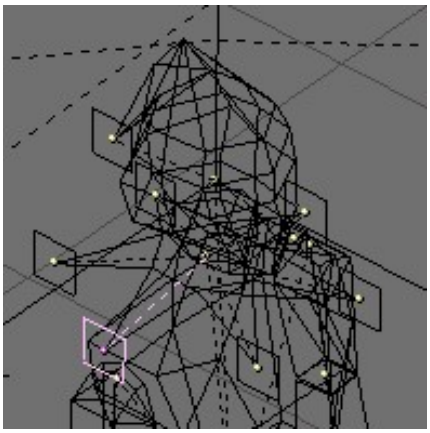


Change to side view (NumPad 3), go out of Edit mode(Tab) and enter Face Select Mode (F). Load "nose.tga" in the ImageWindow. This file is formatted as "RGBA Targa", which means we can use the alpha channel for transparency.

Change the ButtonsWindow to the PaintButtons and press **Halo**, **Alpha**, **Light** and **Twoside** button. Look in the 3d Window (change to "potate-mode" if necessary). Finally change the Button Window to Material Buttons(F6) to attach the material "clown".



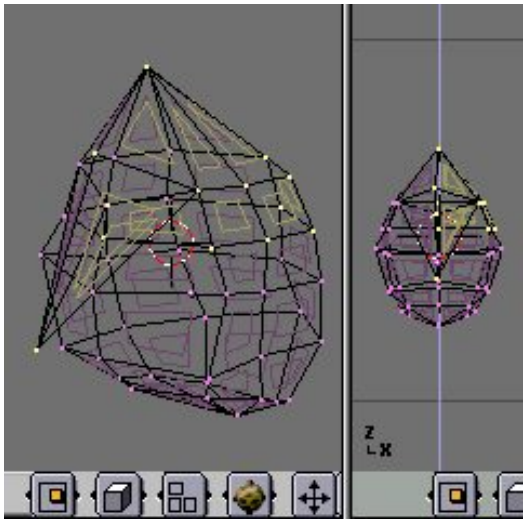
Make the woolen balls in the same way and parent (Ctrl P) them to the "Neck" object.



Vertex Paint

Vertex Paint is easy way to make materials for low polygon models. It is also used to economize on the texture memory and to make static shadows. And you can use it with a texture.

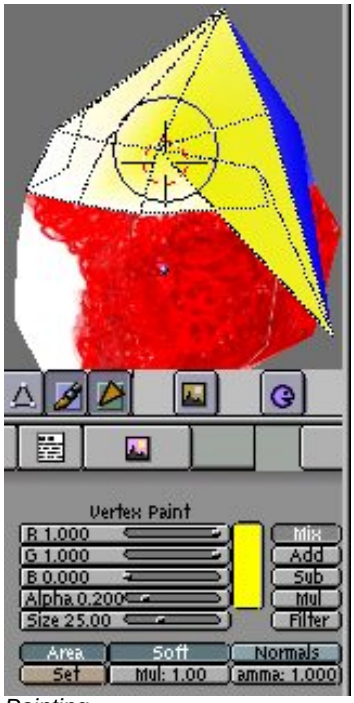
Now, we use Vertex Paint to paint the cap of our clown. Select the head object and enter Face Select mode.



Select target faces

We can select faces in the Face Select Mode with , but we can only select faces that are facing the camera. A nice trick is to select faces in EditMode. Press the Tab key. In EditMode you can see the selected faces (you may have to press 'Draw Faces' in the lower right of the EditButtons F9. Now select the vertices like in this figure.

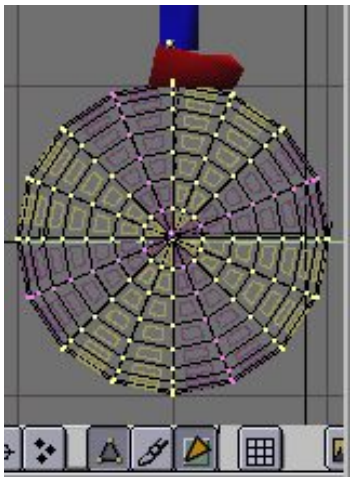
Press Tab key to escape EditMode and open the Specials menu (). Select the **Clr Tex** command to remove the texture from these faces. Change the ButtonsWindow to the Paint buttons.



Painting

Press key to enter Vertex Paint Mode. Set the RGB slide to blue (0.0, 0.0, 1.0) and click to paint faces. When you're finished painting, enter the EditMode again and select the other side of the cap. Now set the RGB slide to Yellow(1.0, 1.0, 0.0) and paint once more.

Now paint the other parts in the same way!



blender

Tutorials

Faking Global Illumination Method

Matt S. - shibbydude

7-18-02

id13

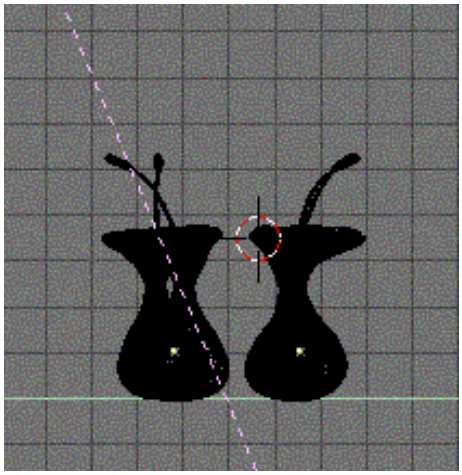
Introduction

Faking Global Illumination in Blender is often tricky for "newbies". When used correctly it gives a soft lighting and soft shadows. This technique gives the benifits and has fewer downsides than "traditional" global illumination techniques. My method is for use only if you can't get the other global illumination methods to work properly (like me for a full year!). Download and open the .blend file below and we'll get started!

Download: GI.blend

Tracking an Empty

When you open GItute.blend you will see four "flowers", a camera, a plane, and a spotlight. I have put the 3D cursor in the middle of the flowers, like this:



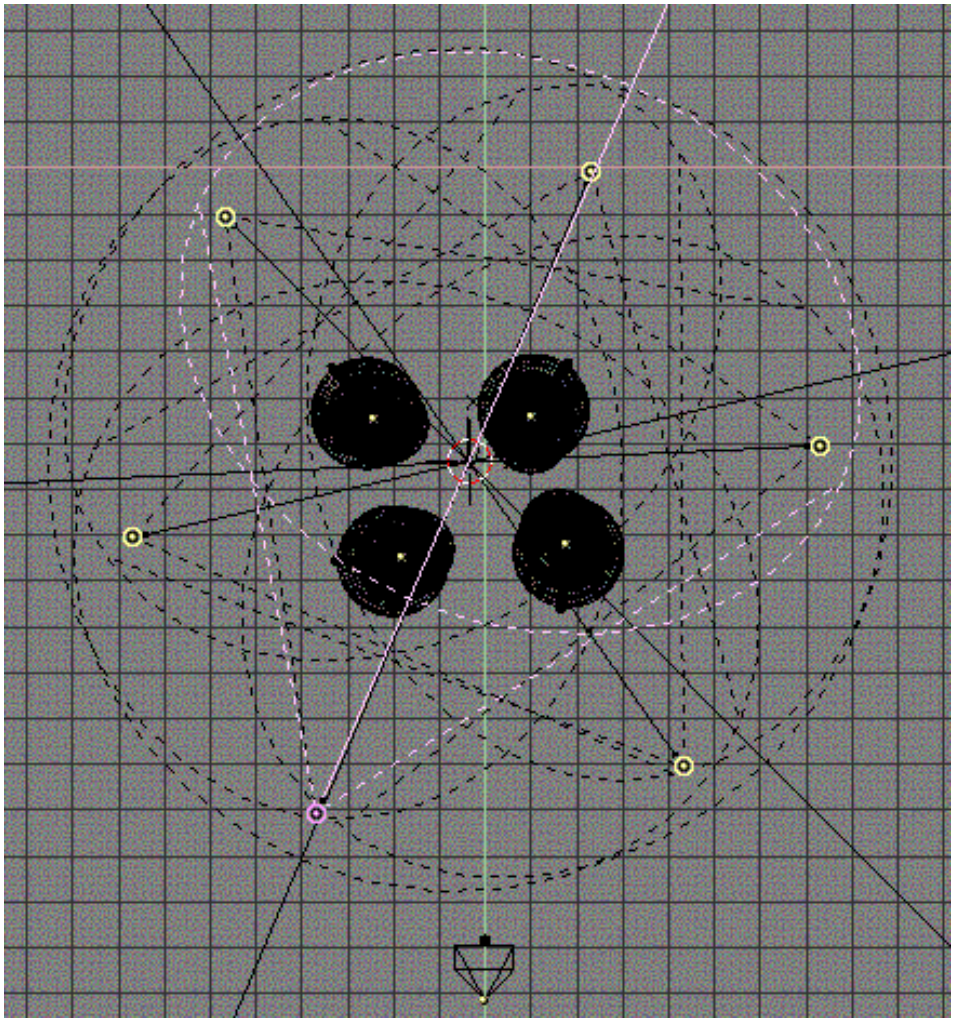
Since the 3D cursor is the location where new objects are placed, hit Space, select "add" and then "empty". You now have an empty at the center of the four plants. Next we will make the spotlight aim towards our new empty using the "track" command. To track the empty, select the spotlight and then the empty in that order and hit Ctrl+T. Now your spotlight should be aimed at the empty. If it is not, select the spotlight and hit Alt+R to clear the spotlight's rotation.

Important:

When tracking one object to another object, always select the target object last.

Duplicating the spots

Now that we are tracking an empty, we need to make duplicates of our spotlight. Hit NumPad 7 to change to the top view. Select the spotlight and hit Shift+D. Right-click and then hit G; move the spot in a circular pattern while holding down Ctrl. You should be able to go up 7 grid spaces and over 3 grid spaces the first time. Repeat this process five times until you have something like this:



Now hit NumPad 3. This gives us the side view. Hit B twice to get the box selection tool. Select all of the spotlights and then hit Shift+D. Right-click once. Now hit G and hold Ctrl while you move the spots down four grid spaces. Now hit NumPad 7. Hit R and then hold Ctrl while you rotate the new spot copies 30 degrees. Now switch back to the side view (NumPad 3) and select all of the spotlights. Hit Shift+D, hold down Ctrl, and move all of the spots down eight grid spaces. We're almost done!

Important:

When rotating an object the number of degrees rotated appears in the lower left-hand corner of the view window.

Linking Lamp data

Our final step is just to save frustration. If we want to change the properties of the spotlights right now, we would need to select each one individualy. To change this simply select all of the spotlights in our scene and hit Ctrl+L. A menu pops open with three options. Select "Lamp data". Now if you want to change the color or intensity of all of the spotlights you only select one and change it - all of the other ones follow it! Now you're done! Hit F12 to render your final scene.

You can download the final product of this tutorial below.

Download: GIFinal.blend

If you had trouble following this tutorial email me at matt@reblended.com. I want to know.

www.blender.nl

this document is online at http://www.blender.nl/showitem.php?id=107

News

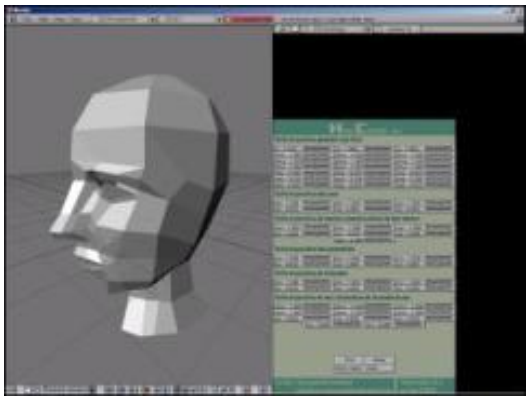
Head Creator v0.3

2000 12 18

Samuel Bourdon

id107

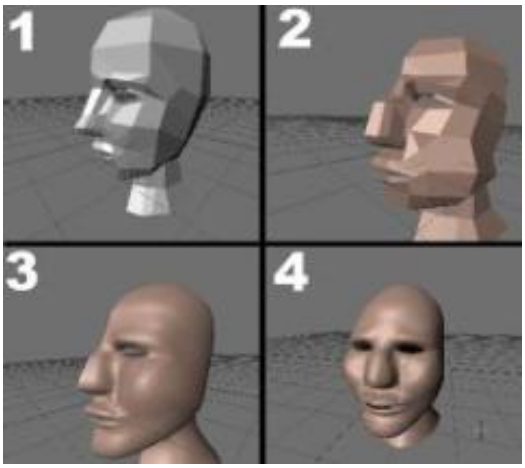
We received this totally cool Python script from SamB. Download hv03en.zip in the download bar below and read the manual here.



French users can find the original text below.

Download: hcv03en.zip

Help for buttons names



Modeling head is often hard and very long. After have staying hours and hours front my PC, trying to make good heads, I've decided to write a python script to make it automatically. I release today the version 0.3

How to use

Download the zip file (english-french) Load the blend file and run the script, the GUI appears. Each slider control a part of the head (Size or Loc). To understand the buttons names print the help file include inside the .zip

Make a head in 4 steps :

1 - Start Blender / Alt+p / Make
2 - Move sliders to have a good looking head
3 - Add some details, active Smesh for better results
4 - Duplicate the half / ctrl +j / remove doubles

When you start the script Head Creator v0.3, you can see that the buttons have strange names. It's just the complete name in shorten.

How to decode them :

Head Creator has two buttons types :

- buttons for the size
- buttons for the position (loc)

All the sliders controling the size are named sx, sy, sz

- sx control the size on the X axis
- sy on the Y axis
- sz on the Z axis

Sliders controling the position are named lx, ly, lz

- lx control the position on the X axis
- ly on the Y axis
- lz on the Z axis

examples:
csx = chin Size X; control the Size of the chin on the X axis mly = mouth Loc Y; control the position of the mouth on the Y axis

General : Size and position / divide in 3 parts :

sx, sy, sz control the general size. The others sliders control the head, divide into 3 parts :

- u == up
- m == middle
- d == down

h means "half"

examples:
hmly = half middle loc y
ndsz = nose down size z

"cdly" slider :

This button means "chin down loc y", and emulate the chin's rotation on the Y axis, moving the lower vertices of it.

"ncy" and "ncz" sliders :

This buttons means "nose curve y,z", they simulate the nose curve on the Y and Z axis. They works like the LOC buttons.

Download: rasta.jpg

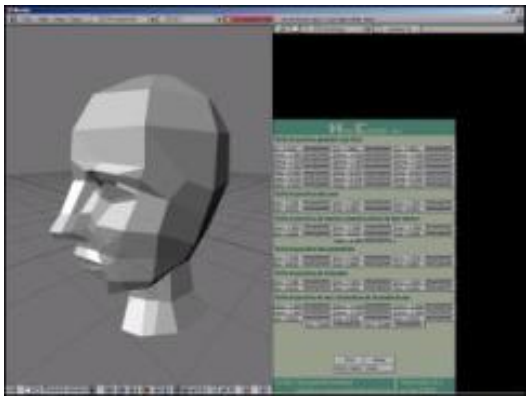
Misc.

Exit : close the script
Make : draw the mesh
Meshname : name of the head (15 carac max)

Have Fun !!!

Sam B 25/11/00

Aide sur les boutons



Lors du démarrage du script Head Creator v0.3, vous pouvez voir que les boutons ont des nomsplutôt incompréhensibles. Ces noms sont simplement les noms complets en abrégé.

Voici donc comment les décoder :

Head Creator dispose de deux types principaux de boutons :

- les boutons controlant la taille (Size)
- les boutons controlant la position (Loc)

Les boutons relatifs à la taille d'un élément du visage auront donc pour terminaison : sx, sy ou sz

- sx controle la taille sur l'axe X
- sy sur l'axe Y
- sz sur l'axe Z

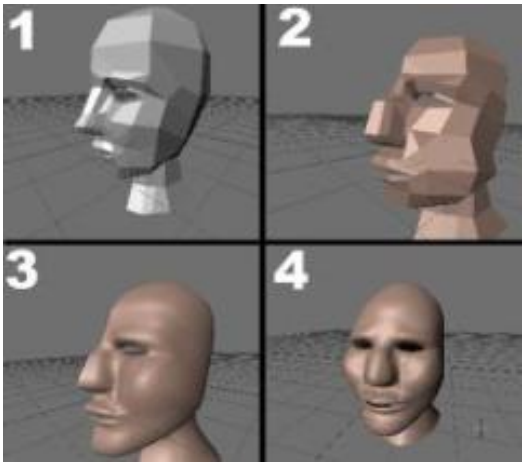
Les boutons relatifs à la position d'un éléments du visage auront pour terminaison : lx, ly ou lz

- lx controle la position sur l'axe X
- ly sur l'axe Y
- lz sur l'axe Z

exemples :
msx = mentonSizeX; controle la taille du menton sur X
bly = boucheLocY; controle la position de la bouche sur Y

Download: hcv03.zip

Exceptions



Concernant les tailles et positions générales "par tiers" :

En plus du controle sur la taille générale du visage (sx,sy,sz),celui-ci a été divisé en 3 :

- h == haut du visage
- m == milieu du visage
- b == bas du visage

d signifie "demi" **exemples :**
dmly = demi milieu loc y
nbsz = nez bas size z

Concernant le boutons "mbly" :

Ce bouton signifie "menton bas loc y", il émule la rotation du menton sur l'axe Y en déplaçant les vertices du bas de celui-ci.

Concernant les boutons ncy et ncz :

Ces boutons signifient "nez coubure y,z", ils simulent la courbure du nez sur les axes Y et Z, ils fonctionnent comme les boutons de type LOC

Download: rasta.jpg

Autres

Exit : sortie du script
Make : dessine le mesh
Meshname : spécifie le nom de sortie du mesh (15 caractères maximum).

Have Fun !!!

Sam B 06/11/00

Feedback

frogus	2000 12 19
very very very cool !!!!	
nathan	2000 12 19
Typical, just spent ages perfecting my hand made head and someone releases a headmaker script! heheh ;) Looks pretty awesome though!	
Brian_Andersen	2000 12 19
Excellent scrpiting work!	
gtman	2000 12 19
very cool, very cool indeed :)	
SiliconGraphics	2000 12 19
I hope this is part of a series, cause this is fabulous!	
Pol	2000 12 19
Félicitations Sam !	
X-tremeBlender	2000 12 18
Lil animation here... lil movement there... talk about some killer cut sense for my RPG! THANX MAN!	
rednelb	2000 12 18
Just what a lot of us were looking for, I guess. Great and many thanks.	
caker	2000 12 18
just what i needed thanks!!!	
assfart	2000 12 18
Oh ah BABY! This head maker is Sweeeeet!!!	
Nick	2000 12 18

a head maker? SWEEEETTT!

heappies 2000 12 18

It is best with Subsurf script!

fish 2000 12 18

WoW!!
Any chance there will be a model script??.
Thanx

FiS}{

ArmyDude 2000 12 18

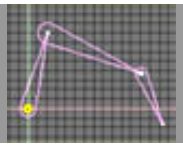
Wayyyyyyy cool - thanks, I needed this

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=139

Tutorials

2001 01 30

IKA basics



Willem Zwarthoed

id139

Introduction

Jeremy Ray sent us this nice Mini-Tute about the basics of IKA. He wrote:

"The Blender 1.8 manual and tutorial seemed to be lacking in their explanation of ika's basics, so I wrote this to fill in the gap."

And now we'd like to share this info with you... Thanks Jeremy!

Have fun!

Zycho

Understanding IKA

Hotkeys:

Shift A - select IKA from the "Add" menu

Esc - Cease adding IKA's. Middle mouse button performs the same function and is quicker.

Tab - Toggle between forward and inverse kinematics. Inverse is the default state.

I - insert key or effector.

Alt A - playback animation (Esc to stop).

→ - advance timeline one frame.

← - Go back one frame on timeline.

Shift → - skip to end of animation.

Shift ← - skip to beginning of animation.

Ctrl K - make skeleton.

Ctrl P - link children to parent objects.

Alt P - clear (delete link to) the parent of a child object.

The Basics

IKA is like the bones in your body. The idea is to animate the "skeleton" of your creature/human, and then let the "skeleton" do the work of deforming your model.

In Blender the skeleton is built from "IKA chains." IKA stands for Inverse Kinematic Animation, which is somewhat misleading as ika chains can also be used for Forward Kinematic Animation. At this point it would be good to know the difference between forward and inverse kinematic animation, right? Think of it in terms of the skeleton inside your own body. When you move your skeleton, it's forward kinematics. If you want to walk, first you move your hip, then you rotate your upper leg, then your lower leg at the knee, then your foot at the ankle, on down to your toes.

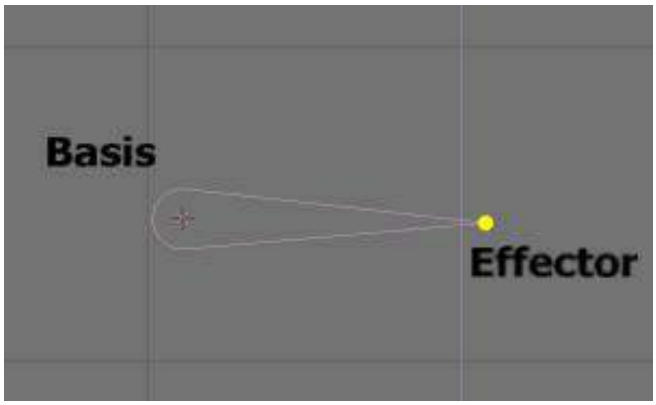
Lots of steps involved, no wonder it took you so long to learn how to walk. You had to figure out how to move all of your joints to put your foot where you wanted it to go - and keep from falling over. Forward kinematics is like learning to walk, you spend a lot of time adjusting IKA's so the foot lands in the right place.

Inverse Kinematics is more advanced, if not without its own quirks. Picture God grabbing ahold of your feet and putting them where they need to go by force. All the other parts of you come along because they're attached to the feet. If you're seeing a guy jerking around funny trying to maintain his balance, you've got an idea of what you're in for animating with IKA (unless you're an experienced animator). It's a strange process to the uninitiated, but it is the quickest way (short of motion capture) to animate your characters.

Let's take a look at skeletal animation in Blender.

The IKA

Here we have a single ika. Notice it has two distinct ends. In Blender the fat end is called the "basis" and the pointed end is the "effector." When using inverse kinematics you will animate the effector. With forward kinematics you will animate the basis. How can you tell which mode you are in? Look for the yellow dot (you must have the ika selected). In the example the yellow dot is on the effector end, meaning this ika is in inverse kinematics mode.

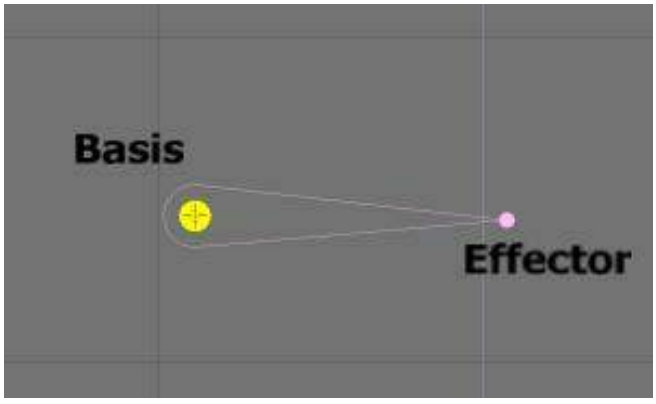


IKA in inverse kinematics mode

In this example the yellow dot is on the basis end. The ika is in forward kinematics mode.

Pay attention to the way Tab is used here. Up to this point Tab has been used to enter/exit edit mode. With ika's it toggles between forward and inverse kinematics, an entirely different function.

Don't confuse the act of selecting forward kinematics mode (starting from the default inverse mode) with entering edit mode elsewhere. You might think either it is necessary to toggle again to leave edit mode or that the ika will revert to inverse kinematics when deselected. This is not so. The tab key, when used on an ika, merely toggles between forward/inverse kinematics. The ika will remain in the state you chose once deselected.



IKA in forward kinematics mode

It's time to start working with ika's. Below you'll find an example blender file to help demonstrate the basics of ika. Load it into Blender now.

At station #1 we have a single solitary ika. Select it (RMB) and note the yellow dot on the effector end indicating inverse kinematics mode. Hit Tab to toggle to forward kinematics mode. The yellow dot moves to the fat end.

You can create your own ika below this one. Keep in mind the ika, like everything else, will be created at the location of the 3D cursor. Use Shift A to bring up the Add menu, and select ika. Place the effector and click out of creation mode with the middle mouse button (or Esc).

Download: IKA.zip

You can draw an ika chain (more than one ika) if you like, but we're going to start looking at animating ika's with a single ika.

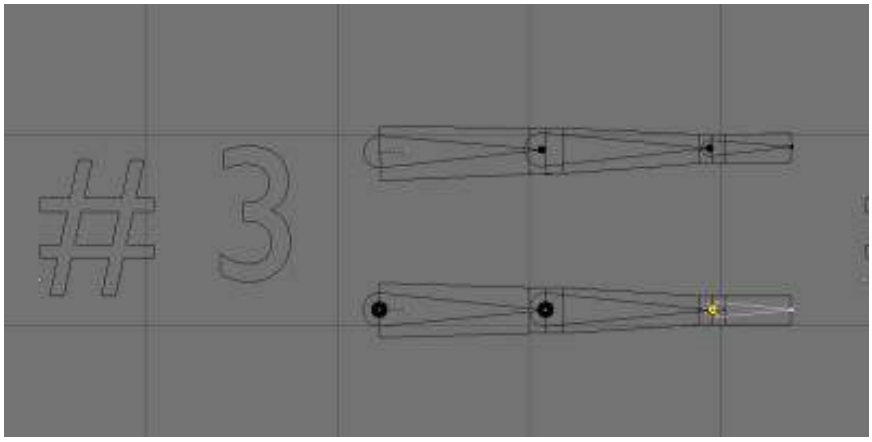
Drag the viewport over to station #2 (Shift RMB). You will see four single ika's. Press Alt A, the hot key for playing animation. You'll see the top two ika's move, but only the topmost one has motion on the basis.

Hit Esc to stop. Now select the top ika. It's in forward kinematics mode. In FK mode only the basis can be animated. You have the option of rotating it (set with "rot" key) or moving it (set with "loc" key).

The second ika is in inverse kinematics mode. IK only gives you the option of grabbing the effector and placing it. You have no control over the position of the basis. IK is animated by setting the "effector" key, not the "loc" or "rot" keys.

The bottom two ika's are for you to experiment with. Set animation keys by pressing I and choosing the type of key. Keys can be overwritten by going to the frame the previous key was set on, repositioning your ika, and resetting the key.

The IKA Chain



Setting up Ika Chains

Now we'll learn how to set up the simple chains from station 3. Pan over to station 4 to create an ik chain and attach it to the simple mesh.

The inverse kinematic chain is by far the easiest to create. Begin the same way as before, Shift A and select ika to begin drawing. Make sure to put the effector in the joint area (you'll see several lines close together). Putting extra faces in the joint area helps the model bend cleanly). Draw from joint to joint until you reach the right most edge of the mesh, where the last effector goes. When you're done it should look like the ik chain from station 3.

Next you have to create a "skeleton" for the ika chain. Select the chain and press Ctrl K. A confirmation window will pop up, click "make skeleton."

The skeleton is a data block, a piece of information not visible within the 3d window. Don't be surprised when nothing seems to happen (but notice the effector turns blue!).

Last step: linking the simple mesh to the ik chain. Select the mesh first and then the chain - the last item selected will be the "active" item. Now parent the mesh to the chain (Ctrl P). An option window will pop up, select "use skeleton." The mesh has been parented to the ik chain. You can deform it as before at station 3.

If you feel you've made a mistake in the linking process, select the child object, press Alt P (clear parent), and the link will be deleted. Try again.

Forward Kinematic Chain

Go to station 5 to set up a forward kinematic chain. Begin creating ika's, but only one at a time. After making an ika, reposition the 3d cursor over it's effector so the basis of the next ika will line up. Then create the next ika until you have all three of them.

Use the Tab key to switch all three of the ika's to forward kinematics mode.

Before you can create the skeleton, it is necessary to link the three ika's together. Start with the rightmost ika, and then select the middle ika (so that it is active). Hit Ctrl P, but this time don't choose "use skeleton" when the option window pops up. Click "use vertex."

Here's what's happening. We want the rightmost ika to be moved by the middle ika. Moreover we want it to be moved by the middle ika's effector and not it's basis, it is necessary to distinguish between the two. After clicking "use vertex" yet another window will pop up, asking you to specify the "vertex" you want. We want vertex no. 1, the effector of the middle ika (vertex 0 being the basis). Click on the right side of the vertex selection button to increment the vertex number. Clicking on the left side will decrement it, in case you go over.

Click o.k. when you've set the vertex to one. Yet another dialogue box will pop up, but we can ignore this one. Simply moving the mouse away from the box will cancel it out.

Repeat the process to link the middle ika to the left ika. Select the middle ika (right ika deselected) then the left ika. Ctrl P, use vertex, choose vertex 1. Move the mouse to cancel out the "effector as child" dialogue.

Make the skeleton by parenting the right and middle ika's to the left ika. Select them in the order right, middle, left. Then press Ctrl K and click "make skeleton."

Finally link the mesh to the ika skeleton. It is only necessary to select the mesh and the parent (left) ika, then Ctrl P and choose "use skeleton."

And that's it, the mesh is ready to be animated! Using these types of bones you can make an entire skeleton to deform a complete character. Unfortunately that is beyond the scope of this Mini-Tute, but we hope it'll get you going...

Feedback

nozzy	2001 01 31
nicely done!	
zou	2001 01 31
finally !	
Toonami_fan	2001 01 31
good job.	

<MATT>

RipSting	2001 01 31

This is a good tutorial, but IKA skeletons are what really get me!

Baron_M 2001 01 30

to Ansi :
you have some Particle Emitter BASICS on http://www.blender-cafe.org

About this tutorial : a cool one :)

Angel7 2001 01 30

Interesting and Informative :p

Wish this had been there a year ago :)

cool!!

Melvil 2001 01 30

Nice. It will be very useful when I finnaly get into character animation....

Pusher 2001 01 30

Usefull!
It will help me a lot for my game posted in the
http://blenderbypusher.s5.com/blender!

calli 2001 01 30

I just need to say that there are IKA Basics in the Tutorial guides and a very advanced IKA Tutorial in the "Official Blender 2.0 guide".

Particle emitter basics are also in there.... (from very basic to dupliverted, latticed schoal of fish)

Carsten.

mrunion 2001 01 30

Thanx!

Ansi 2001 01 30

nice !
finally somebody explains the basics.
would be nice to get an tutorial about Particle Emitter BASICS.

keep up the good work!

www.blender.nl
this document is online at http://www.blender.nl/showitem?id=12

Jump!

Bart Veldhuizen

Tutorials
2000 07 24
id12

Introduction

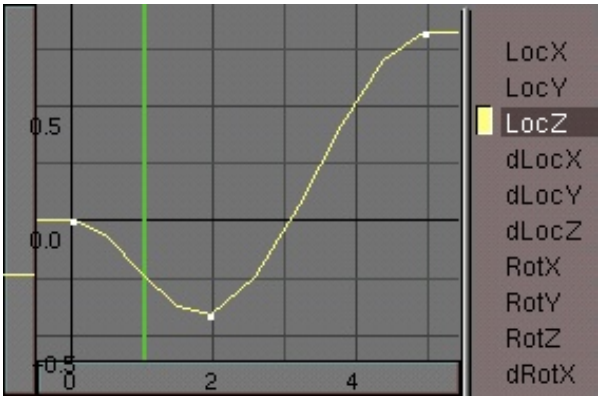
Now that you know how to control your character, it is time to make it do something a bit more interesting. In this short tutorial, I will explain you how to make your character jump.

Force IPOs

I want to make my character jump by applying a force in the vertical direction to it. This force should only last a few frames. To make the jump motion more realistic, I will first apply a small force downwards (causing my character to 'crouch' and prepare for the jump). Immediately after that, an upward force will be applied.

You already know how to use keyboard sensors. Select your character and add a new sensor for the Enter key.

Now to create the variable force. Split your 3D window and open an IPO screen with Shift F6. With your character still selected, select the LocZ channel in the IPO window. Hold down Ctrl and draw the following IPO curve:

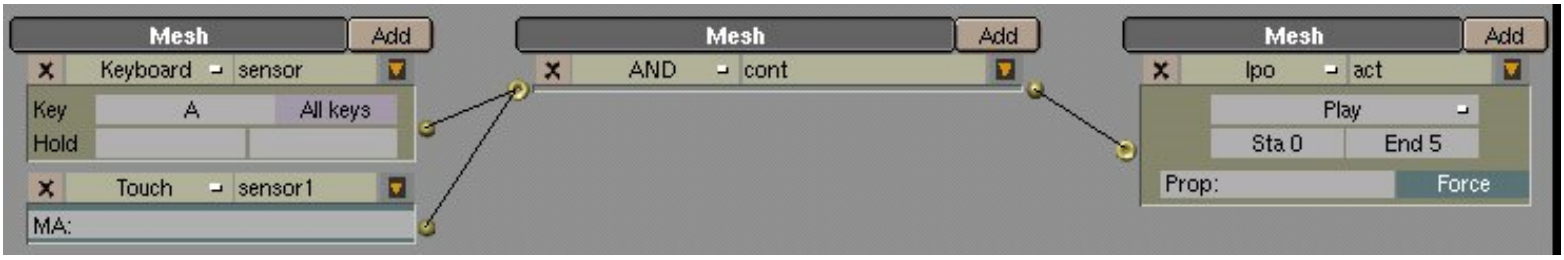


Force IPO curve.

Right now we have a location IPO, but I want to apply it to my character as a force. First, select your character and turn it into a dynamic object (select Actor, then Dynamic in the Real-time buttons window. Add an IPO actuator and connect it (using an AND controller to the keyboard sensor. Keep the default IPO type setting of 'Play'. Set the start end end frame to the correct value (0 and 5 in my case).

To make the IPO work as a force, check 'Force'. Activate your simulation with P and jump by pressing Enter.

IPO force setting.



Jump logic.

Something is still not right with the jump: if you jump in mid-air, the force will be applied to your character. Of course, you should only be able to jump while you touch the floor.

This is easily solved by adding a touch sensor to your character. By default, this returns a TRUE when the character touches something. Anything will do (walls, floor etc).

Connect the output of this sensor to the same AND controller as the keyboard sensor. The logic now requires that your character jumps only when you press the jump button AND it is touching the floor.

Before this works, there is still one more thing that we need to do. Remember the Fh Dist value? The cushion area around an object not only determines when the Fh force starts to work on other objects, but also when collisions are registered. If you leave the Fh Dist value of the floor at zero, collisions will not register at all.

Select the floor object and turn to the Material Buttons window. Select 'DYN' and increase the Fh Dist value. Play with your simulation until you reach a good value: a higher value makes it easier to jump. At the same time, it can make the movement more unnatural.

If you only want to detect a touch event with certain objects, you can enter a material name in the sensor's MA: field. Only touches with objects that have this material will return true now. You can download my example file here:

Download: jump.zip

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=154

Tutorials
2001 03 01

Particles

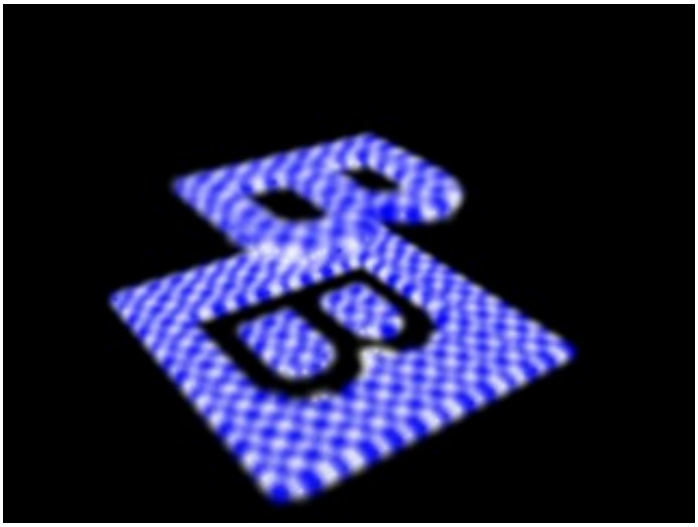
Predrag Kurtovic

id154

Introduction

Particles are a nice feature of Blender. You can use them in any segment of animation. Blend files with particles are small because particles are of a procedural nature, but you need more computer power to preview the scene in real-time. You can give every family of particles a different material, make fireworks, use static particles for hair and plants and give them a shadow. I want to teach you how to command particles where to go. This unique Blender feature use the eighth texture channel to calculate the particles movement. I will divide this tutorial in two parts: in the first part I will tell you how to get "solid" particle movement, and in the second how to make particles go in the turbulence using texture.

Solid particle movement

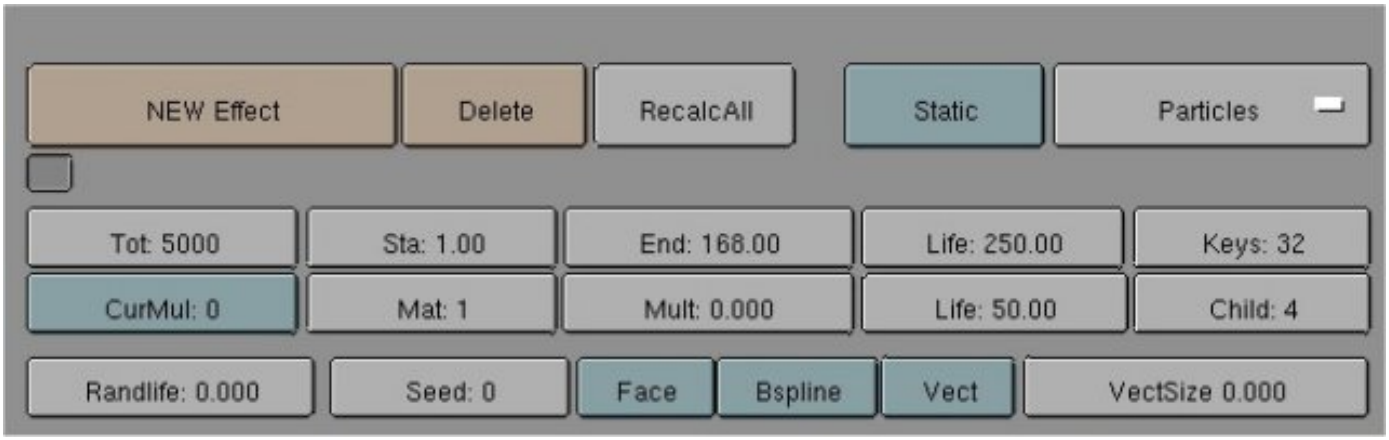


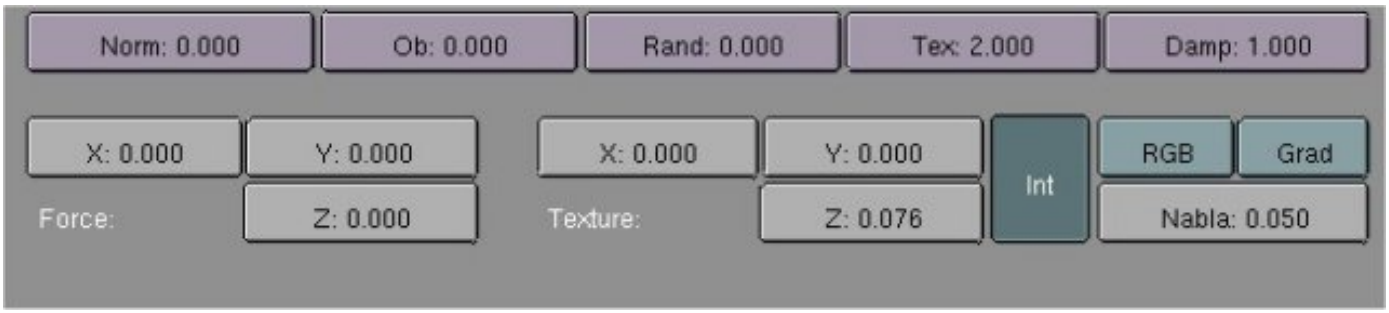
First of all you need a texture. Make it in any graphic program like Gimp or Photoshop. Let it be black texture with big white letter "B" in the middle. This picture will "tell" the particles to follow the shape of letter, which creates a very interesting effect. In the default scene apply a new material to the Plane and use this picture as texture (it's packed in the blend file).

Download: controled.zip

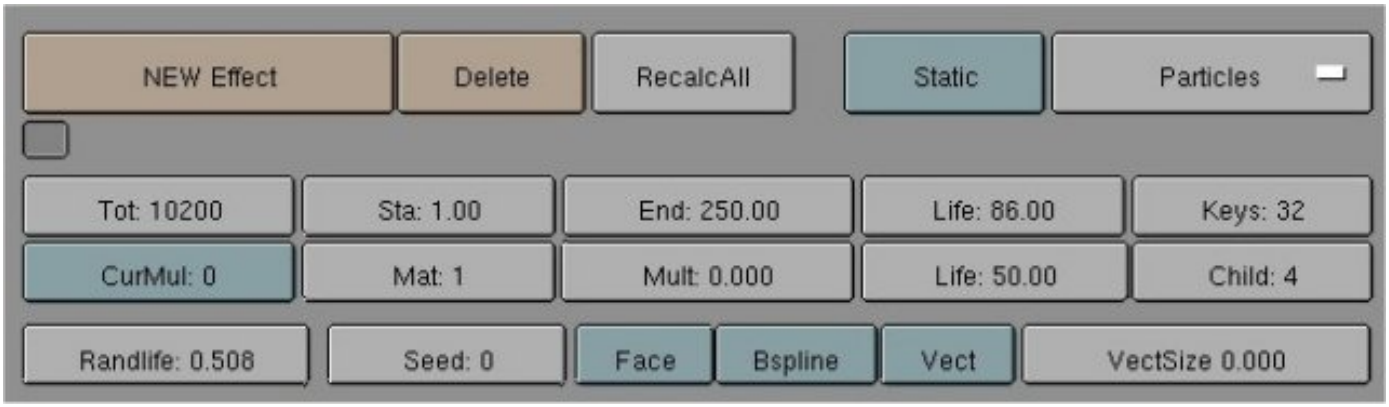
There is one important thing: **you must use this picture in the eighth texture channel (the right most channel in the MaterialButtons), because this is the only channel that can give this effect. Because of this the eighth texture channel will not be used for texturing particles, but of course you can still use the other channels to texture the particles.** Do a few subdivides on the plane to get more emitters so you can get a more accurate picture of the shape you create.

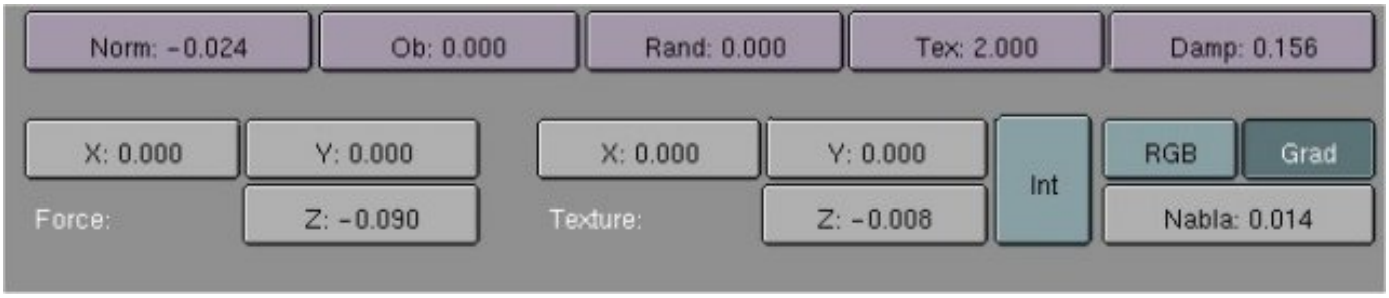
Solid particle movement





Dancing particles





A very important thing is setting Blenders animation system. You have the option to choose between: Int ,RGB and Grad. Int is default settings for particle systems. In option RGB Blender calculate speed and movement of particles according color of texture. Red is for x component of speed, Green for Y and Blue for Z component. In this tutorial we use last option: Grad. "Grad" means the gradient of the texture. With help of mathematics Blender calculates the speed vector from the texture. Composed with high a high keys value this gives beautiful twisting effect. But be careful, you must make many adjustments to find the "right" value for nice twisting! Nabla is set to 0.014. Nabla button select dimension of texture area where the gradient is calculated.All you need is to apply some material to particles (snow texture) and animate it!

Download:	itsnowing.mpg
Download:	itsnowing.zip

Final words...

I hope that this tutorial will teach you how to control particle movement via texture. Any suggestion or question can be sent to e-mail Happy Blendering! Predrag

Feedback

Ed_2021 — 2001 03 05

oh yea... i also would like to know how to play my animation in RealPlayer, or Windows Media Player. Or can i watch them in Blender?? Tell me how Please!!!!

Ed_2021 — 2001 03 05

I would like to know how to make my own screensavers in blender. Is it possible??

lingesh — 2001 03 05

i wish i can print this tuttorial.....but noooo.....all i get is ascii....

thorax — 2001 03 05

Texture controlled particles has been in blender
for the longest time, texture colored particles has not.

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=122

Tutorials
2001 01 09

Radiosity

Matthew R. Duncan

id122

The final result

Introduction

There has been a lot of buzz recently in the channels about a technique called Global Illumination. In my first tutorial, I hope to explain this technique a little better and give you a good starting point from which to try your own GI project.

Traditional still scene rendering usually involves lamps, spots, etc. But GI is a radiosity technique that allows you to light a scene from 360 degrees without a single lamp. This gives your scene more of a stylistic look.

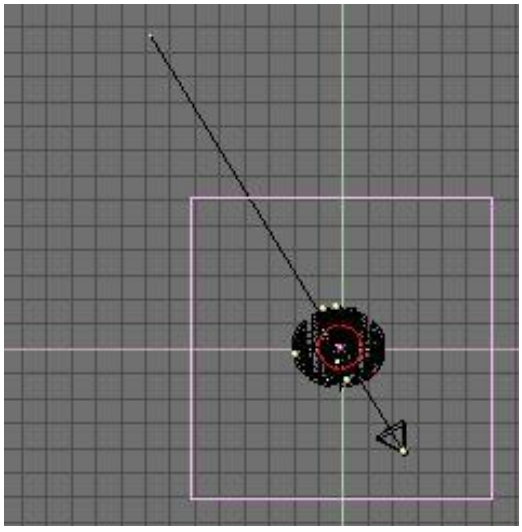
This image is an example of what can be done with Global Illumination. This tutorial will teach you how to work with radiosity and how to apply textures after the radiosity process is finished. I assume that you are already familiar with creating meshes, applying materials, and have a general understanding of the Blender interface. While you can certainly use your own scene for this tutorial, I have provided my Raider mesh below as a starting point.

Download: GI.zip

Setting up the Scene

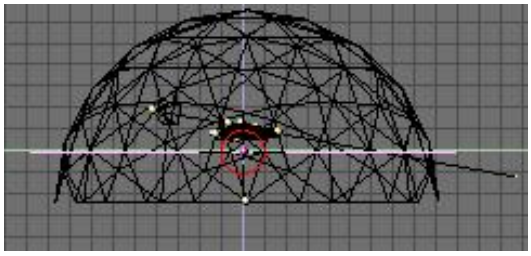
In the .blend file provided, you will notice that there are only two elements in the scene: a Cylon Raider and a camera. The Raider has the default grey material, except for the main cockpit windows which are black. For this technique, we will not need any lamps.

The first thing that we will want to add to the scene is a plane. This plane will be used as the floor in our scene. Resize the plane as shown on the right and place it just under the Raider. Leave a little space between the plane and the Raider bottom. This will give us a nice "floating" look.



Add a plane

Next, you will want to give the plane a material and select a color for it. In my images I try to use a nice blue. You can use the setting on the right for this.



The dome

The Dome

The next thing that we are going to add is an icosphere. This sphere is going to be our light source instead of the typical lamps. What we are going to do is use the vertices as "emitters" that will project light for us in multiple directions instead of in one direction as with a typical lamp. This will give us the desired effect.

To set this up, add an icosphere with a subdivision of 3. While still in EditMode, use the B key to select the lower portion of the sphere and delete it. This will leave us with our dome. Resize the dome to better fit the scene and match it up with your plane. It should resemble the image to the right.

Next, we want to make sure that we have all the vertices of the dome selected and then click on the EditButtons F9 and select <DRAW NORMALS>. This allows us to see in which direction the vertices are "emitting". Then select <FLIP NORMALS>, which will change the vertex emitter from projecting outward to projecting inward in our dome.

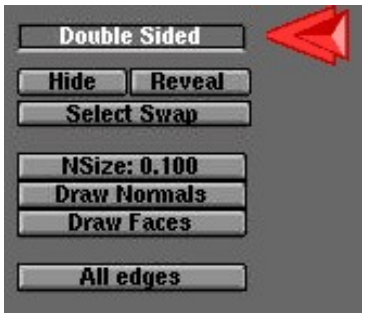
The Dome Material

Now that we have created our dome, we need a new material. When you create the material for the dome change the following settings in the MaterialButtons F5:

Material settings for the dome

Add - 0.000
Ref - 1.000
Alpha - 1.000
Emit - 0.020

The Emit slider here is the key. This setting controls the amount of light "emitted" from our dome. I generally use 0.020 but you can experiment with this setting to get different results. The lower the setting here though the longer the "solve" time later.

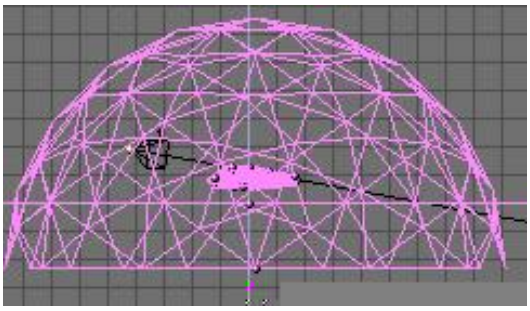


Converting Meshes

At this point we have created everything that we need for our scene. The next step will be to alter the dome and the plane from "double-sided" to "single-sided". To achieve this, we will select the dome mesh and then go back to the EditButtons F9. Click the <Double Sided> button and turn it off. Repeat this process for the Plane.

Collecting Meshes

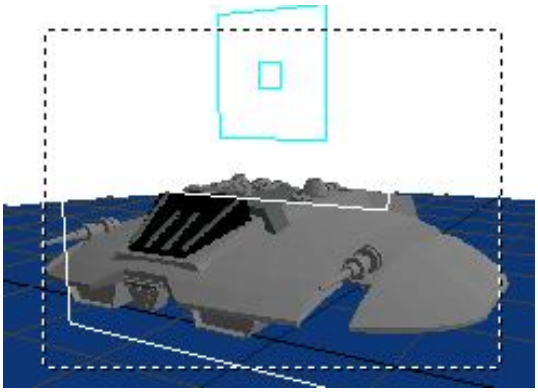
Now the next few steps are the heart and soul of Global Illumination. Go to side view with NumPad 3 and use the B key select all of the meshes in our scene. Next hold Shift and double click on your camera. We do not want this selected. It should look similar to the image to right.



Select all the meshes

After selecting the meshes, go to camera view with NumPad 0 and then turn on shaded mode with Z so we can see inside our dome.

Now select the RadiosityButtons. On the left-hand side of the menu, click the "collect meshes" button. You should notice a change in your view in the colors. It should look similar to the image to the right.

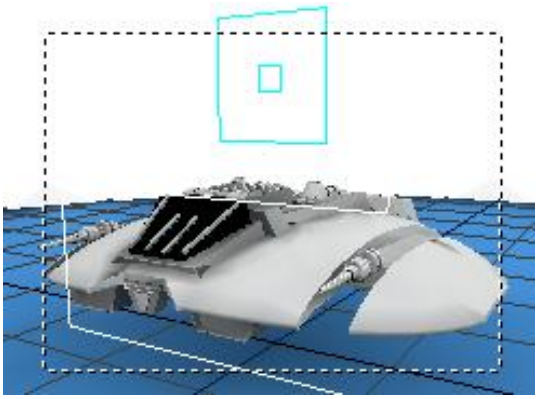


The RadiosityButtons

Next, to keep the Raider smooth like our original mesh, we will want to change from "Solid" to "Gour" (#1). This will give our Raider its nice curves back, in the same way "Set Smooth" would in the EditButtons F9. You'll also need to change the "Max Subdiv Shoot" to 1 (#2). Do not forget this step.

"Solving" the Meshes

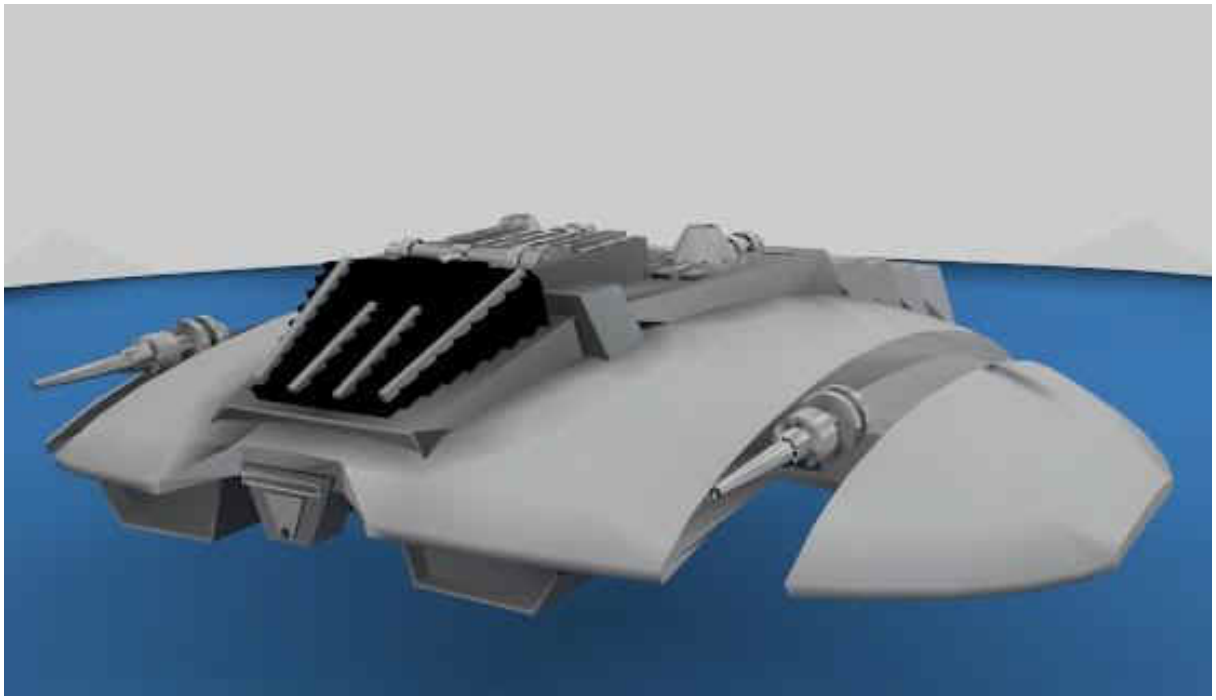
After you have set "Gour" and "Max Subdiv Shoot", click "Go" and wait. Blender will then begin calculating the emit part of the dome, going face by face, thus "solving" the render. As it does this, you will see the scene change as more and more light is added to the scene and the meshes are changed. You will also notice that the cursor in Blender changes to a counter much like if it were an animation. Let Blender run, solving the radiosity problem.



I usually let Blender go to somewhere between 50-500 depending on the scene. The solving time depends on you and how long you decide to let it run...you can hit Esc at any time to stop the "solve." This is an area that can be experimented with for different results. This can take from 5 to 10 minutes and your system speed will also greatly determine how long this process takes. The picture above is our Raider after 100.

After hitting the Esc key and stopping the "solve," click "Replace Meshes" and then "Free Radio Data." This finalizes our solve and replaces our previous scene with the new solved radiosity scene. It should look like this...

Now we are ready for F12 and render...and viola!

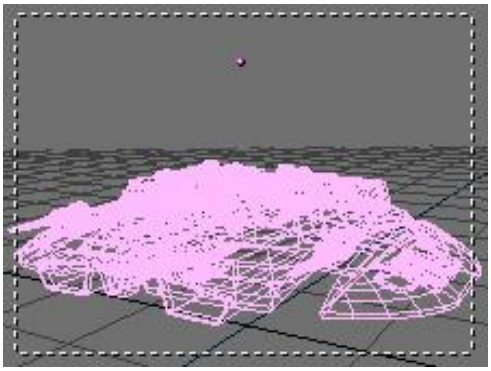


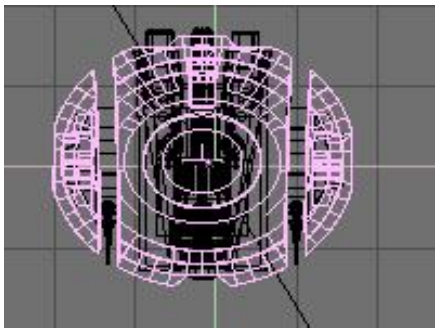
Moving on to Textures

There you go folks! You now have a very clean looking render with soft 360 degree lighting using radiosity. Very nice... But the next thing we want to do is add textures to the mesh. So hit Esc to exit the render and go back to our main screen area.

Now try selecting your mesh and you will notice that it selects not only the Raider but the plane and dome as well. That is because we created a new mesh through the radiosity solve. To add a texture though, we only want the Raider.

So, select the mesh and then go into EditMode. In EditMode we can delete the dome and plane since they are no longer needed. You can use the B key to select the proper vertices and press X to delete them. Keep selecting and deleting until you are left with only the Raider. It should look like the screenshot. If we were to render it now with F12, we would get just a black background and our Raider. This is nice.. but again, we want textures!





Separate parts with 'p'

Splitting the Mesh

To add textures to mesh, we must separate out the areas that we are going to apply materials and textures to. For the Raider, I wanted to add textures to the wings and mid-section. To do this select the Raider mesh, and go back into EditMode. Select a vertex near the edge of the wing and then hit the L key to select linked vertices. Do the same on the other side. Next, click on the mid section of the ship and do the same thing. Select the areas shown in the figure. When you have those, hit the P key to separate the vertices selected.

Adding the Material

We now have our wing section separate and are ready to add the materials and textures. You will want to create a new material for this mesh. To get a nice metallic look, I have used the settings you see in the figure.

A metallic material

Time to add the textures. There are four textures for the Raider wings and these textures can be downloaded below. The next step is adding them to the material.

Download: textures.zip

Adding the Textures

To add the textures go to the TextureButtons F6, select new texture and then click on the image button and load the texture. You will need to do this with each of the textures, assigning them to the same material. You should have the following textures (see figure).

Negative NOR and REF setting

The textures should be set up as follows in the MaterialButtons F5: RaiderBM.JPG and RaiderDI.jpg should be set to a negative NOR (click NOR twice, it will turn yellow). Raider. JPG should be set up as negative REF.

The result is the desired metallic plating for the hull of the Raider. Finally, markings3.jpg is set to COL in the MaterialButton F5. This will give the Raider its proper striping and insignia. The material preview for the mesh should look like the image.

Light the Scene and Render

For our textures to show up in our rendering, we will now need to add a lamp or two, keep in mind that our ship is still lit pretty well from the radiosity solve so tone down your lamps energy. Once you have your lamps, you try a test render with F12. Experiment with the lamps until you get the best results.

Here is the final rendering... A nice well lit Raider with soft texturing. If you would like a copy of the final blend, you can get it below.

I hope you have had fun with this tutorial and will enjoy this new way of using Blender and radiosity!

Download: GI_final.zip

Feedback

WileCoyote	2001 01 09

xlent!!! Thanks for the visual aids! and the textures. . .

vogelap	2001 01 09

I especially like the little arrows indicating steps in the screenshots. VERY handy, especially when the item is printed!

Mad-Sci	2001 01 09

Great job Cujo, it took a while to get out there, but it was well worth the wait.

Extremely helpful information, finally there is a good tutorial for this method of global illumination. I am sure you will help a lot of people with this. Keep up the great work!

-Dereck G.

MatrixNAN	2001 01 09

I love it great idea!

Nate Nesler

Kib_Tph	2001 01 09

very nice Cujo31

Pol	2001 01 09

Such a cool tut !!!!!!

gdi12	2001 01 09

This is very cool! It reakky help me a lot!

www.blender.nl
this document is online at http://www.blender.nl/showitem?id=99

Tutorials
2000 12 05

Refraction

Willem Zwarthoed

id99

Materials in general

First I want to start off by saying a little something about materials in general. A good material, in my opinion, is a material, which makes a model appear believable, or realistic. I love it when I'm looking at an animation and I get the feeling I can really touch the models. Creating such 'good' materials can be pretty tough, since Blender offers you a great amount of control over a materials properties.It takes a lot of tweaking to get your material to look *just* right and you shouldn't expect perfect results at the push of a button. Getting to know Blender's settings and the way different variables interact with each other is crucial, but it's also very important to have a keen eye for the real world. You'll never be able to make a realistic looking material just by tweaking the many variables if you don't study the appearance of the object you're trying to recreate.

Also note that it's the animator's objective to create effects that appear to be realistic or believable, not effects that are physically correct! This holds true for almost everything in 3D animation. But of course that doesn't mean that some knowledge about the real world doesn't help.

One other thing: there's no such thing as a perfect material. A wood material may look very nice in one scene, but might look horrible in another. A materials settings need to be adjusted according to the rest of the scene, according to lighting and surrounding materials (this holds true especially for the refraction technique discussed in this tutorial). It's very important to set up good lighting for your materials, because materials interact with Blenders lights.

TIP: When experimenting with different settings it can be very useful to compare renderings. Luckily Blender has two render buffers: Render your image. Hide the render window and change your settings. Press J to toggle the render buffer. Render the second image and now you can toggle between the before/after images with J. You can also press Z to zoom on images to examine artifacts.

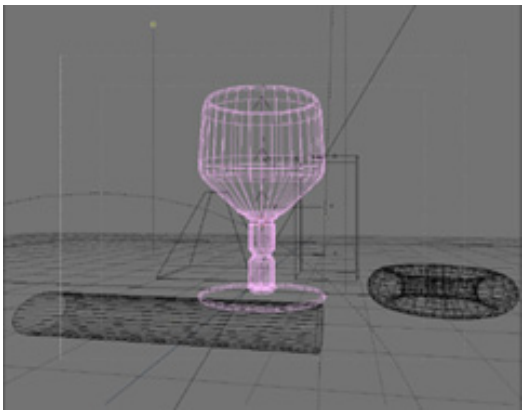
Okay, so much for the philosophical part. Let's get to the fun stuff! This tutorial is about making realistic looking glass, so I've spent a lot of time examining glass (or it's appearance, actually) closely - and I have to say the bartender from the bar around the corner has started to like me a lot... Taking a closer look at the glasses there I noticed a few key points, of which most importantly that glass is never truly transparent.

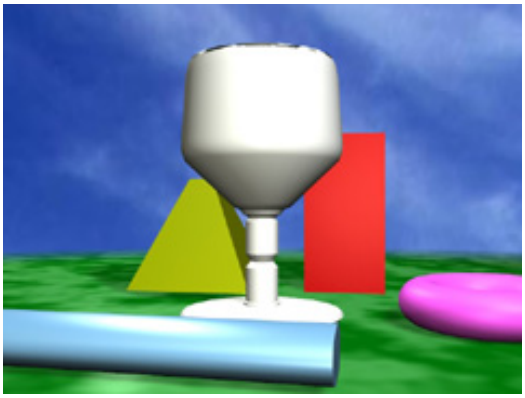
This means that glass:
- refracts it's environment,
- reflects it's environment,
- has imperfections.

The basic settings

I've set up a little scene for you, so download the file 'glass.blend' and load it into Blender. You'll see the scene I set up, containing a few basic objects and the glass.

Download: glass.zip



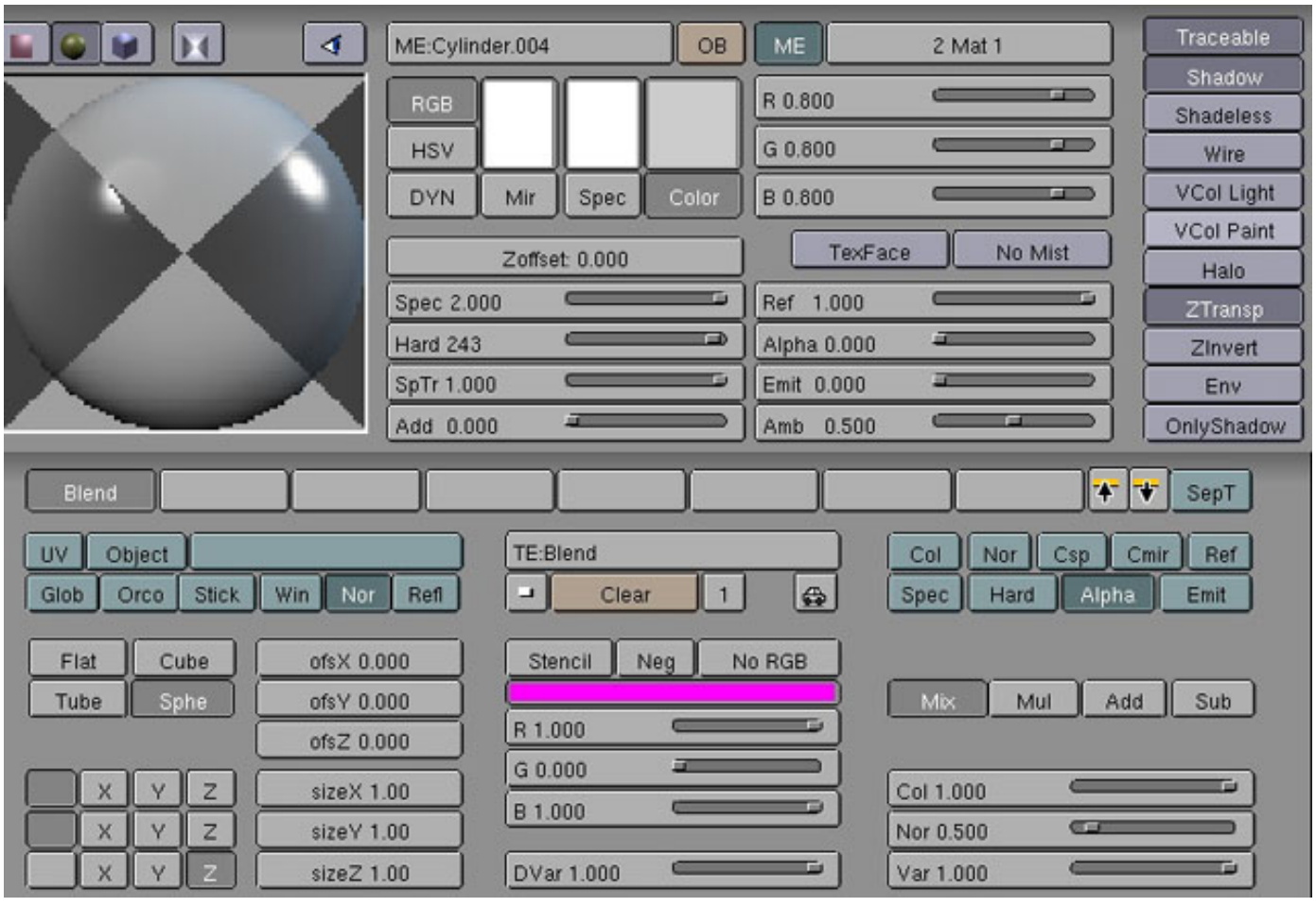


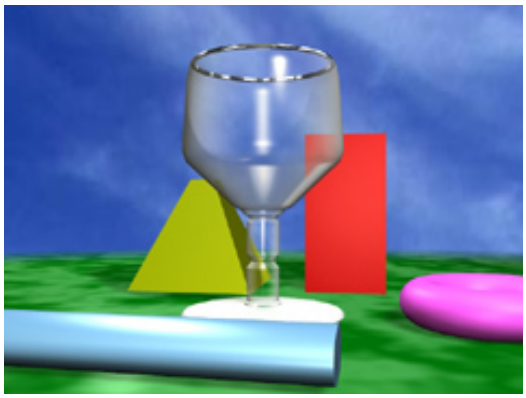
There are a few things to take notice of in the scene.
First the glass I modeled has no inner wall. I removed the inner wall afterwards because the final result looked much better without it (if you ever tried making a glass material with Ztransp turned on you probably know you always see the inner wall as well and you can't see the inner wall like that with a real glass).

The second thing is the glass has two material indices, but only one has a material assigned to it. If you do a test render you'll see the second material index is the chrome edge of the glass and that the rest of the glass has the default material, which Blender uses to render (parts of) objects without a material assigned to it.

Okay, time to get to work. We'll start by assigning a new material to the glass. With the first material index selected, select 'Add new' in the MaterialButtons F5 and name it 'Glass'. Set Spec, Hard, SpTr, and Ref to full. A 'simple' way to fake glass can be found in the file 'tutor_1.6.zip' available from ftp://ftp.blender.nl/pub/.

Let's use this as a starting point: add a spherical blend texture with a spherical nor mapping. Disable all mapping except for the Z-axis. Select 'Mix' and 'Alpha' for the output mapping. You'll have to turn down the materials alpha and select Ztransp to see the effect. If this is a load of nonsense for you take a look at the screenshot for the settings.





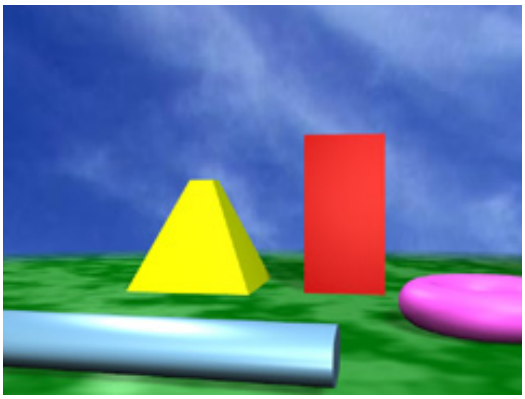
If you render the image you'll see the glass is starting to look better, but we're far from done. After all we wanted to have refraction.

A suggestion offered for this is to use an environment map with a negative SizeZ value. This can turn out pretty nice as long as you're working with spheres or other basic shapes, but when the geometry becomes more complex the fiddling with the settings becomes very cumbersome.

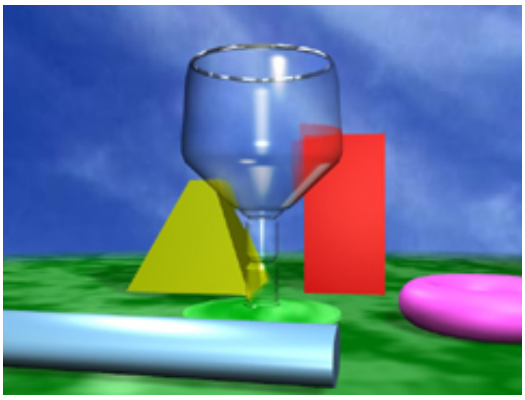
Think about this (I did, anyway): what does refraction really mean here? The way I see it, refraction means that anything behind the glass should be visible, but it should be distorted according to the glasses geometry and imperfections. When you're trying to use inverted environment maps you're seeing the objects behind the glass from the glasses perspective, which results in very extreme distortions not suitable for thin, hollow glass like we're trying to create here. What I really want to see is the scene without the glass and to distort that image according to the glasses geometry and imperfections.

Refraction

That sounds like a nice idea, so let's render the image without the glass. Move the glass to layer two using M > 2 > Enter and hit F12 to render. Press F3 and save the image as 'refmap1.jpg', add a new texture to the glass material and name it 'RefMap'. Choose image in the TextureButtons F6 and load the image we just saved. Press Shift+2 (mouse over the 3D Window!) to reveal layer 2 with the glass and render the image to see what it looks like.

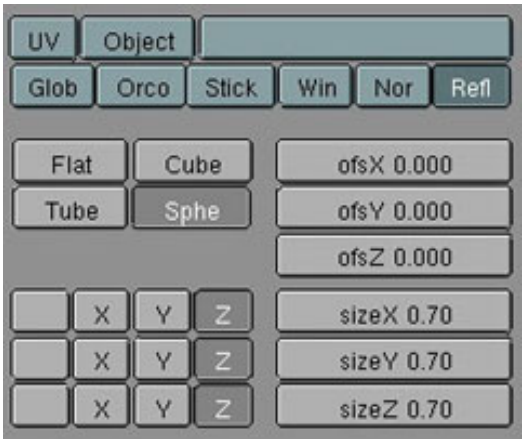


refmap.jpg



It's not much yet is it? If you play around with the different mapping types, you'll notice that finding the right look appears somewhat impossible, with the exception of the 'win' mapping. However this mapping is undesirable because it doesn't allow much deforming. After a good afternoon of tweaking the different settings I finally found the solution. Use the camera for the texture coordinates! The only things that need adjustment are the SizeX and SizeY parameters, which relate to the distance between the camera and the refracting object. This is no real problem for stills, but when animating you'll have to keyframe these parameters. Perhaps it would be nice to write a python script to link the Size to the distance, like Iceman's script that automatically places the empty for planar reflections.

Set SizeX and Y to 0.40 for now, and it should look okay. Well... maybe not, but at least the refraction map looks okay.



My glass doesn't look like glass yet!

Now we've got the refraction set up, but the whole thing does not look like glass yet. One thing that doesn't look right is this blend texture. Like I said, making a material look good requires much tweaking, and finally I found the following settings look best for this glass.

Another thing that needs to be taken care of is the shading on the glass. If you look at glass closely, you'll notice that it doesn't receive any shadows. The first thing we can do to fix that is to click the button labeled 'Shadows' which controls whether or not an object is able to receive shadows from other objects. Though it looks better this doesn't solve it completely, because the shading from the lamps is still visible.

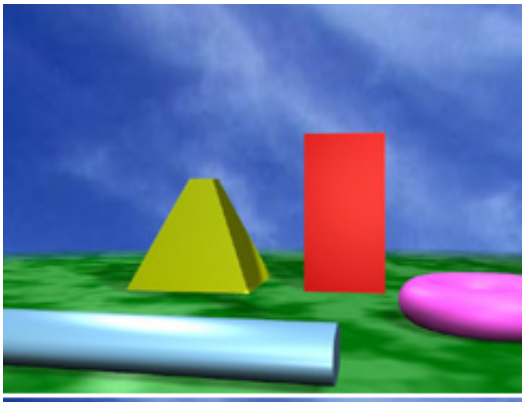
Now we don't want to set the 'shadeless' option, because then we'd lose the specularity as well. A quick way to fix this is to set 'emit' as output mapping for the refraction map. This, however, is a setting that makes your glass very sensitive to its surrounding materials!

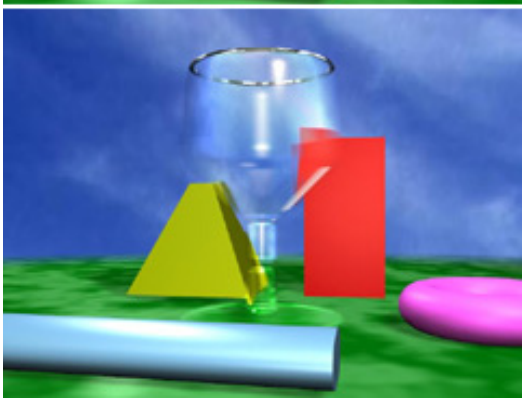
If you think the refraction looks too bright you can adjust the amount of 'emit' with the 'Var' slider.

We're still not done, so what else is wrong? You might have noticed that the yellow pyramid doesn't receive a shadow from the glass in the refraction map. To fix this we'll need to render the refraction map again, but this time we'll need to do it with the shadow from the glass.

Here goes: duplicate the glass and move it to layer 3. Name it something like 'Glass.Shadow'. Add a new material to it and name it 'Shadow'. We only want the shadow, so this duplicate has to be invisible. The problem is that if you set the Alpha to 0 and click Ztransp it wont cast a shadow anymore! A little trick to solve this is to set the Alpha to 0.001 instead of 0.000. That way it's still (practically) invisible, but shadows are still cast. Now go to the EditButtons F9 and delete the second material index. Make sure '2 Mat: 2' is selected (if you're using my file it'll say 'Chrome' above this button) and click the pink button labeled 'Delete'.

Now render our new refraction map by selecting layers 1 and 3 and hitting F12. Save the new map as refmap2.jpg or overwrite the old one. Disable layer 3 and enable layer 2 again. Select the glass and reload the RefMap image in the TextureButtons F6. If you did everything correctly up to now you should have a pretty glassy looking glass now.





The little things...

Wow! Almost done... now it's time to add two more details (I mentioned them earlier): reflection and imperfection!

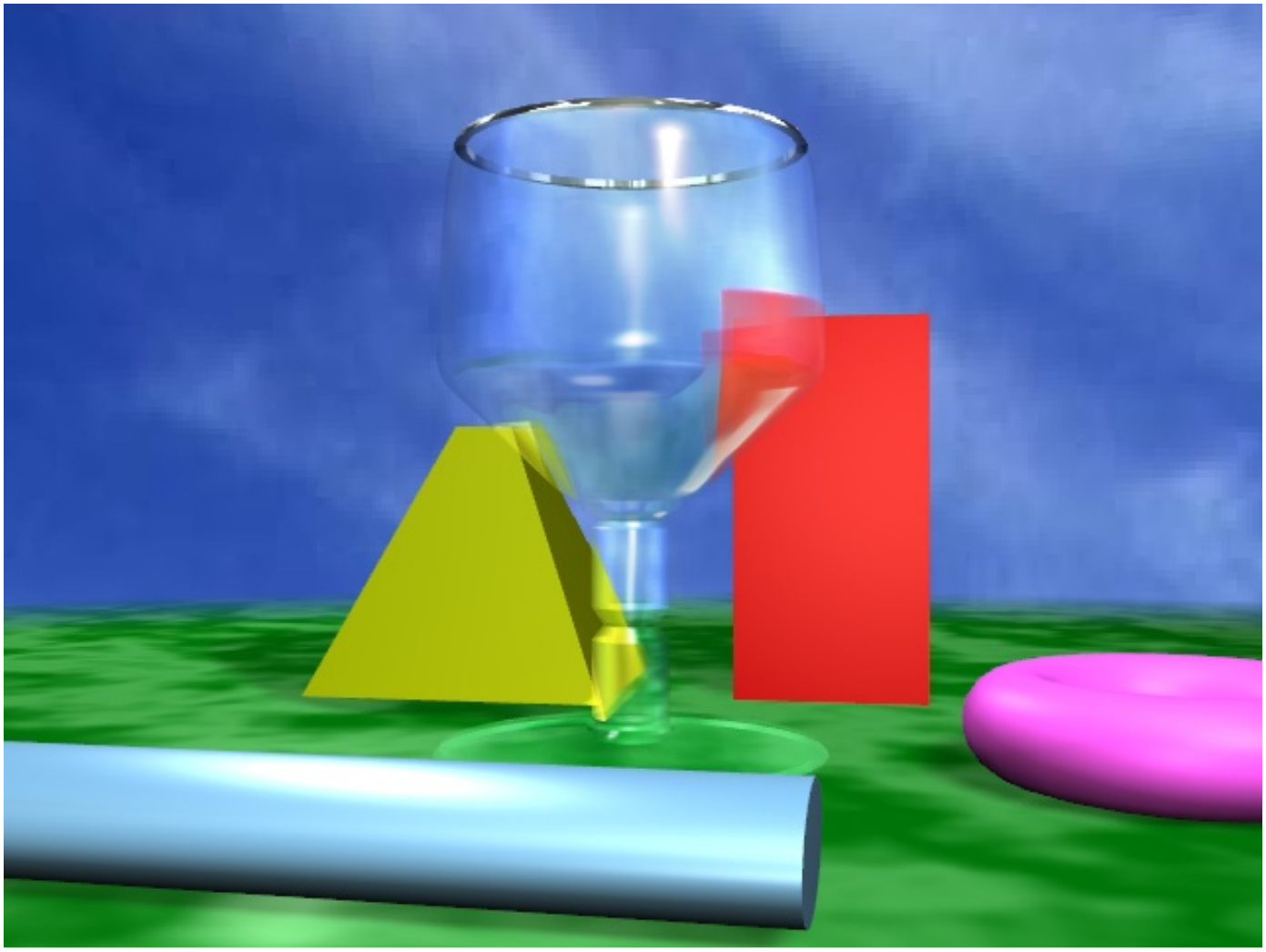
The imperfection will be done with a stucci texture. But to make it work the stucci texture needs to be on texture channel 1 (a texture can only affect textures in higher channels), so we'll need to move the other textures up. Now you're probably thinking 'Oh no! Now I have to memorize all those little settings and enter them again and I don't wanna!' Luckily Blender has a copy/paste buffer for this sort of thing, which makes it pretty simple to move textures around in the different channels.

Back in the MaterialButtons F5, select the blend texture. To the right of the texture channels you can see two buttons with arrows on them. The one with the up arrow is the copy button. Click it and Blender informs you with a popup, 'Copied!' Click it or move the mouse away. Click the pink button labeled 'Clear', select channel 2 and paste the blend texture with the button with the down arrow on it.

Now move the RefMap texture to channel 4 (!) and add a new texture in channel 1. Choose stucci in the TextureButtons F6. Choose 'nor' as the output mapping in the MaterialButtons F5. You might want to play with the NoiseSize and Turbulence, but for now I left them at the default setting.

Now all that remains is the reflection.

Add a new texture in channel 3 and choose envmap and enter 'Glass' in the Ob: field. To get a nice and clear reflection the cube res has to be quit high and filter should be low, but that increases the render time... At a resolution of 640x480 I found cube res at 250 and filter at 0,10 (the lowest setting) looked best. Select Col, Alpha and Cmir as output mapping and set the Col and Var sliders to 0,250. Finally you might want to play a bit with the Offset of the reflection to make it look more natural. I found OffsY: 0.100 and OffsZ: 1.200 looked nice.



In closing

Pfew! That's quite a tutorial (to write, anyway). Render the scene and marvel at your glass... you might want to try the Unified Renderer, which gives stronger highlights and clearer transparency (rendering magic!).

A little note on animation: of course you can animate the refraction map as well, but you'll have to render the entire animation without the glass first. You'll have to adjust the Clipsta value of the camera so any objects in front of the glass won't be rendered. Load the animation as a texture and don't forget to set the number of frames. If you saved the animation as AVI you'll also need to click the green movie button. You can download the animation below. You can also download the finished .blend file.

A little note on caustics: if you want the glass to really look realistic then you should also add caustics to it. Unfortunately this is beyond the scope of this tutorial, so you'll have to find a nice way to do this yourself (or maybe I'll write another tutorial, who knows!).

Of course you can always send me an email if you get stuck or have any questions or suggestions. Don't hesitate to contact me at: willem@blender.nl

I hope you enjoyed this tutorial,

Zycho

Download: glass_done.zip

Download: glass.mpeg

Feedback	
dmx	2000 12 06
Finally i can i can make the glass for my F-14 Tomcat!!!	
jaf	2000 12 05
VERY USEFUL! THANX	
WCFireball	2000 12 05
Nice! Now I can finaly complete my house! (Up to now it was without windows!)	
Taylor	2000 12 05
let there be glass!! finaly i can do good glass.:-)	

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=57

Tutorials
2000 11 09
id57

Particle System Shadows

Evan Weaver

Introduction

See *Blender Tutorial Guide 2, Chapter 8: Static Particles*. Those plants don't cast shadows, and that's pretty dumb-looking when we're aiming for photorealism. But by careful use of image textures and mapping, we can work around that.

By rendering an image texture from the perspective of the light source, changing it to a form of black and white, and subtractively mapping it to the objects that are to receive the shadows, we can achieve a very realistic effect (See Image 0).

Image 0: What we can accomplish

Step 0: Getting Started

What you need:

- A scene with a particle system. (Grab one off the Tutor Guide 2 CD-ROM if you have it, otherwise you can download it below. Of course the ideal way would be to make your own.

- An object you want it to cast shadows onto.

- A spotlight, set up to cast shadows normally (your non-particle system objects' shadows render as usual).

- The GIMP, the 2D equivalent of Blender, because it's also very free (actually, The GIMP was first, so Blender is the 3D equivalent of it, I guess.) You can use a similar program like Paint Shop Pro or Photoshop if you are comfortable with it, but this tutorial uses The GIMP.

- Not to be upset by someone who starts counting at zero instead of one. :)

Download: begin.blend

Image 1: begin.blend before shadowing

Steps 1-5: Rendering the shadow image

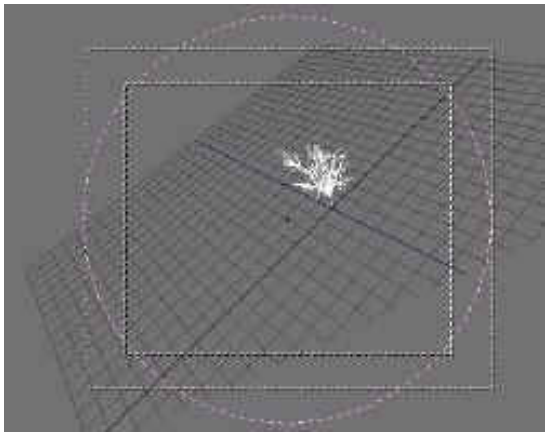


Image 2: The view from the spotlight

1. Move your spotlight and your particle system to their own separate layer, because we don't want to render any of the background right now. (Right click on each object, M, 2, click OK)

2. Move to this layer only (2), so all that is displayed is the lamp and the plant.

3. Select the spotlight, and press Ctrl keypad 0. The spotlight becomes the current camera (see Image 2).

4. Go to render buttons, change **sky** to **premul** (unless you don't have a world object defined), and render the current layer at the final resolution. (Premul renders the background as black instead of whatever skies you have set up.)

5. Save the image as something like *fake_shadow.tga*.

Steps 5a-5f: Altering the shadow image

5a. Open the image in The GIMP. (There is a Windows version now, so you have no excuse.)

5b. **Right-click** (on the picture), **Image**, **Colors** (depending on GIMP version), **Grayscale**.

5c. **Right-click, Image, Colors, Curves**. Adjust the curve so it is at the very bottom on the right, but after about 10 pixels shoots up to the very top. Turn on **"Preview"**. Notice the image is basically black and white now, except, everything that was not black is white, instead of normal black and white conversion (see Image 3). The idea is to make leave the black background completely black (0 0 0), but make the particle systems completely white (255 255 255). So, a dark gray (50 50 50) would become white instead of black like it would if you converted it to two color normally. See Image 3 for "curve" settings, and 4 for the intended result.

5d. Repeat (c) a couple of times until the image is only black and white.

5f. **Right-click, File, Save**.

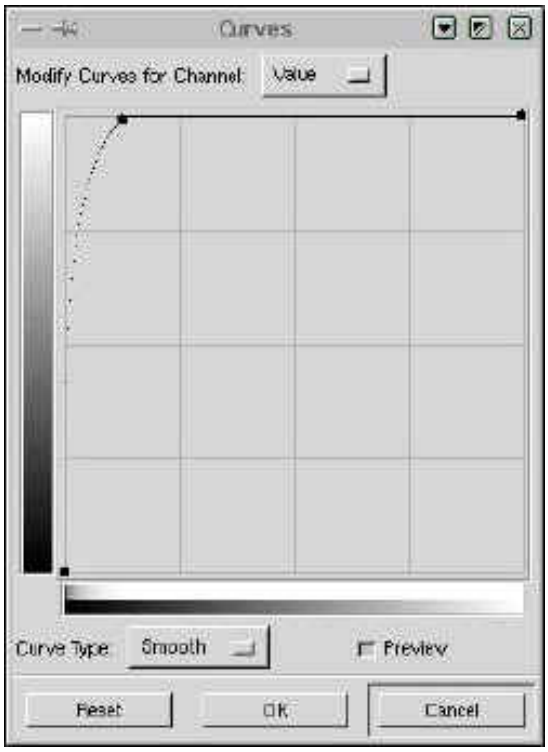


Image 3: Your "curves" window should look like this

Steps 6-8: Some miscellaneous actions

6. Press **tilde** (the ~ under escape) to activate all layers again.

7. Hit Alt keypad 0. The view returns to the camera.

8. Select the floor, wall, or whatever you want to cast shadows onto.

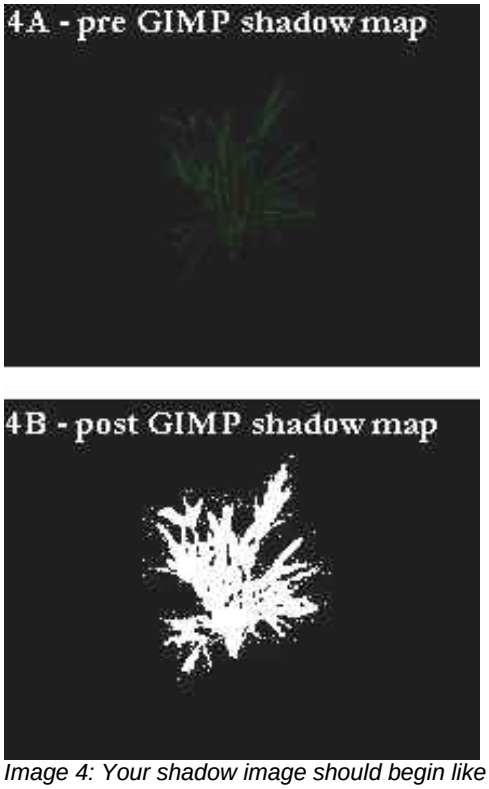


Image 4: Your shadow image should begin like 4A but look like 4B after "curving"

Steps 9-11: Creating the texture

9. Create a new texture in the floor's materials. Set it to image, set the filter to around 4 (adjust later). Load fake_shadow.tga as the texture image. Turn on **UseAlpha**, and **CalcAlpha**. (This makes it so the places where the shadow isn't don't affect the rest of the object.)

This texture must be last in the texture order, so that the shadows get put on top of everything else. (Unless, of course, you want to be difficult and fool with **stencil** and other oddities.)

Image 5: The material settings

10. Go back to the material buttons. Set the new texture's color percentage slider to 0.500 (Above **alpha** slider and **var** slider). This means the material will blend the texture half and half with the underlining colors and textures. Change **mix** to **sub** (subtractive instead of blending). Bright spots on the shadow texture are subtracted from the object, causing dark areas, or shadows. (This method avoids white edges on the shadow caused by the filtering process. Took me a while to figure that out...)

11. Set the new textures **size x**, **size y** and **size z** all to 0.100 or so. (This value is the *inverse* of the distance of the SpotLamp to the objects creating the shadows. Inverse means one divided by the number; i.e. for a lamp 10 units away, 1/10 = 0.100). *And here's the key* - instead of the default **orco** mapping type, set the mapping to **object** and type in the name of your SpotLamp (probably **Lamp** or **Lamp.001**; you should change it to something descriptive.) This calculates the texture depending on the position of the lamp.

This procedure must be done for each object that is to receive a shadow, but the texture they all use should be the same (*don't* "make single user"). All the **size** and mapping (**object**) values should be the same in every material belonging to objects that are to receive shadows.

12. Render the entire scene (F12).

13. Be amazed.

14. Go back and adjust parameters marked "adjust later," and keep repeating steps 12 and 13 until the output looks correct.

15. Save.

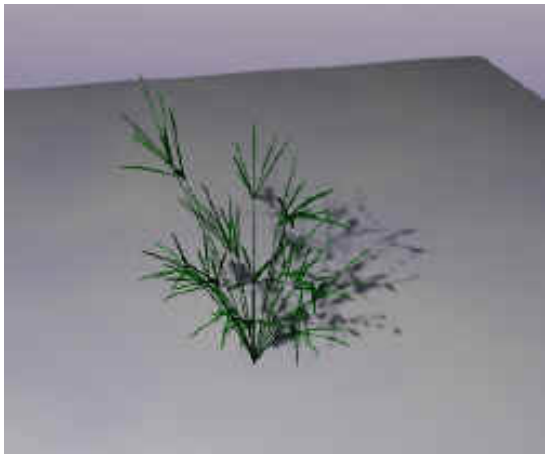


Image 6: The final result

And there we are, a particle system with beautiful shadows!

This procedure works only for stills. You could make it work with movies, too, using an animated texture, but that's beyond the scope of this tutorial. And personally I will probably never do it, because my work is all for printing. It's tough to print an animation.

Note for expert users: If you have a ton of objects, adjusting every value in every material is painful. You may want to consider using straight-line ipokeys, and linking each separate material's value to the same ipokey. Then you could simply move the ipokey up or down to alter the value everywhere. This could save you a lot of work in the long run. But you've got to know what your doing...

You can download the final result below.

Here's a shameless plug for the author's **Blender website**, where you can see a high-resolution version of Image 0, among other things.

Download: final.blend

Feedback

eweaver 2001 02 11

In reply to Sk8er_27:

In a word: No.

In reply to Dale:

Blender Tutor Guide #2, available in the Blender Shop, explains all about it.

Sk8er_27 2001 01 18

I dont Like doing my work in 2 different programs. is there any way to do it all in blender?

Dale 2000 12 07

Cool, but how do you do the plant itself??
When I start playing with particles, it ends up pretty messy =)

MATE 2000 11 29

Thanks for tutorial...Am now heading over to see your web site

Mad-Sci 2000 11 11

Amazing tutorial! Thanks a lot, I will deffinately be putting this one to good use.

-Dereck G. (a.k.a. Mad-Sci)

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=128

Tutorials
2001 01 12
id128

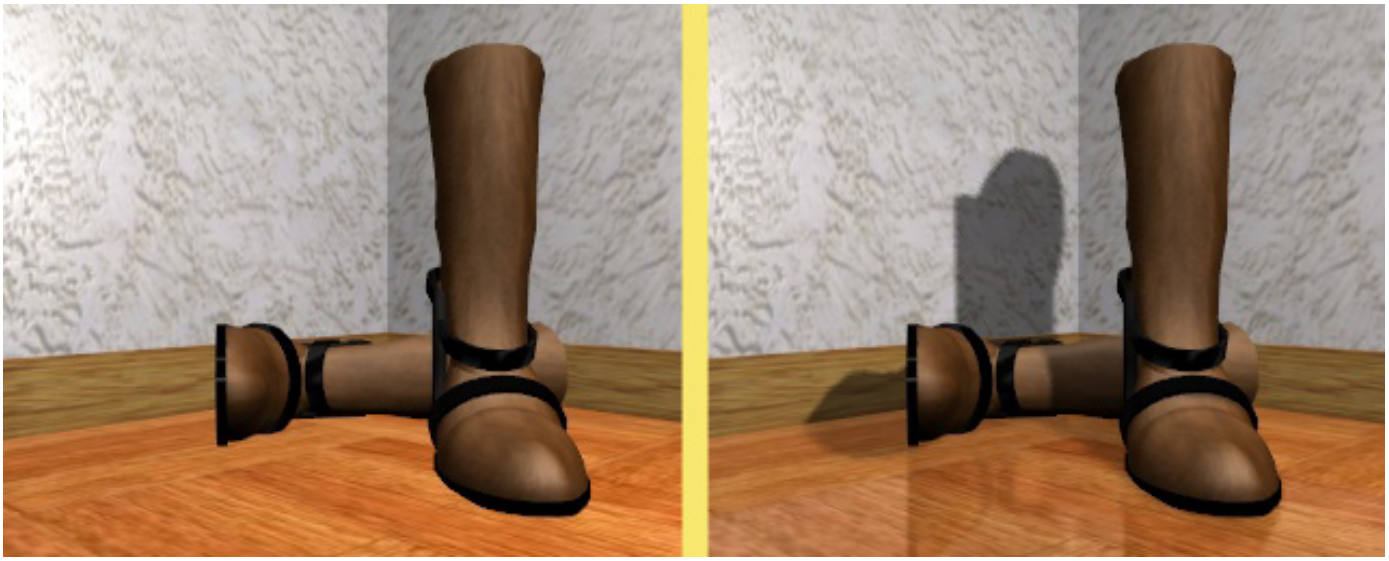
The Spotlight

Olivier Saraja

Introduction

Beginners are often puzzled during their very first renderings. They usually think that they only have to create a model and give it a material for the picture to be rendered, forgetting that a light source, like a lamp, is necessary in order to get anything else than a black screen. Once this very first detail is mastered, they surely wonder why there aren't any shadows in their scene.

This tutorial is all about generating shadows (which is pretty easy) and making them look the way they should (which can become very intricate).



A cool beginner picture... transfigured by the use of EnvMap on the floor and, of course, the use of shadows!

Note about this scene : the topic of the scene is somehow inspired from the outstanding photorealistic researches led by Manu Batôt on his own webpage . The boots model comes from Michael's Blender Website.

Casting shadows

There are a few pre-requisites for Blender before being able to cast shadows. They won't be reminded anymore during this tutorial.

In the Material Buttons F5:

- Materials which should cast a shadow must be set with the "Shadow" button 'on' (default in Blender). By enabling this button, Blender calculates the shadow cast by this object.
- Materials which should receive the shadow of another object should have the "Shadeless" button set 'off' (default in Blender). By enabling this button, Blender won't cast shadow over this object.

As these are defaults Blender values, you shouldn't have to bother with them at first.

In the Lamp Buttons F4:

- The only lamp type than can be used for casting shadows is the Spot.

- The Spot lamp should have the "Shadows" button set to 'on' (default in Blender). The Spot then cast lights AND shadows.
- The Spot lamp could be set to "Only Shadow" if there's another lamp in your scene. The Spot then ONLY cast shadows, but no light.

The default Blender lamp is the Lamp, which doesn't cast shadow, so you HAVE to set at least one light source as a Spot if you want to cast shadows. Please note that among all lamp types available in Blender, the Spot is the ONLY one that can cast shadows.

In the Render Buttons F10:

- The "Shadows" button should be set to 'on'.

The default value for this button in Blender is 'off', so don't forget about it if you want to cast shadows!

Basic knowledge

In order to calculate shadows, Blender uses the shadow buffer algorithm. This means that the scene is rendered from the point of view of the Spot, and that Blender calculates which object is placed before the other on its line of sight, thus defining which pixels, 'hidden' from the spots view by anyone object, will receive shadow in the final render.

The way it operates could be explained as follows, even if it is far from being accurate and involves different mechanics (like storing a pixel distance from the Spot, for example): in our example picture, here's a render from the Spot point of view of what the shadow cast by the foreground boot could look like. Given that the light (white) values are discarded, this picture could be 'mapped' on the objects located behind the boot, with the Spot origin as coordinates for the texture, with the shadow color adding to the base color of the object.

Of course, if there are many Spots in the scene, Blender 'renders' the scene as many times, from each Spot point of view, and stores the results in its shadow buffer in order to bestow light or shadow values accordingly to each pixel of the final render. This is why multiple Spots add to the rendering time.

With this in mind, it is now easier to understand the use of some of the instrumental buttons and sliders in the Lamp Buttons F4. In fact, the parameters we will deal with hereafter are set individually for all the Spots of a given scene.

Size of the shadow buffer (BufSi)

As we have seen before, information about pixels that get light or shadow is stored in a buffer, which acts exactly the same as a rendered picture. This means that the shadow buffer has its own resolution, than can be set accordingly to the effects your are after. By default, Blender's buffer size value is 512x512 pixels.

BufSI set to 512; Zoom: The shadow is blocky

This kind of effect is good enough when the camera from which the final picture is rendered isn't too close to the shadows, because at a distance it is very hard to tell about the blocky look of the shadow.

BufSi set to 2048; Zoom: The shadow is far more progressive

When you increase the Buffer Size, of course, the shadow gets more accurate borders, which is better for close shots of an object and its shadow. BufSi behaves exactly as the resolution of a rendered picture does

Size of the Spot (SpotSi)

The angle of beam of your spot also has a great effect on the blocky look of your shadows. I rendered the previous pictures with a SpotSi set to 120.00 degrees.

BufSi set to 512, but SpotSi set to 120.00 degrees; Zoom: As seen previously, the shadow is blocky

Now let see how well the rendering does with a smaller beam, set to 60.00 degrees.

BufSi 512, SpotSi 60.00; Zoom: The shadow is more progressive, but less than with a higher BufSi

It is clear enough that the smaller the angle of beam is, the sharper the borders of the shadows are. There's an immediate conclusion to this: before setting the BufSi to any great value, first enhance the quality of your shadows by seting the SpotSi value as small as possible ! You will spare both memory and rendering time ! We could say that SpotSi behaves as the lens of a camera does.

Samples

In the previous renderings (BufSi set to 512, SpotSi set to 60.00 degrees) we can see that the shadows are smooth, but perhaps not enough. Before trying to increase the BufSi, we will try to set the number of samples to a higher value.

BufSi 512, SpotSi 60.00 and Samples 3; Zoom: The shadow is quite acceptable

BufSi 512, SpotSi 60.00 and Samples 5; Zoom: The borders have been anti-aliased

It is very hard to tell the difference between Samples 3 and Samples 5 at first, but looking at the zoom closely, it becomes obvious that the borders of the shadows have been smoothed. In fact, Samples behaves the way OSA does.

Intermediate conclusion

You have now achieved a better understanding of how shadows should be cast in order for them to be the highest quality as possible. You have understood that BufSi is the key to quality, but also the cruder mean to achieve it, because it asks for a lot of memory. As a thumb of rule, let say the following :

- Try to set all your SpotSi within 45.00 and 60.00 degrees values.
- Set Samples to 5. Try to set BufSi to 768 or 1024.

With this in mind, about 70% of your renders will have the best quality/render-time ratio. If you have enough RAM memory and if time isn't a factor for you (eg, you are dealing with a still picture, not an animation), then you can set the BufSi to the highest value you choose. A high BufSi combined to low SpotSi values and high Samples gives, of course, the very best results Blender can let you achieve.

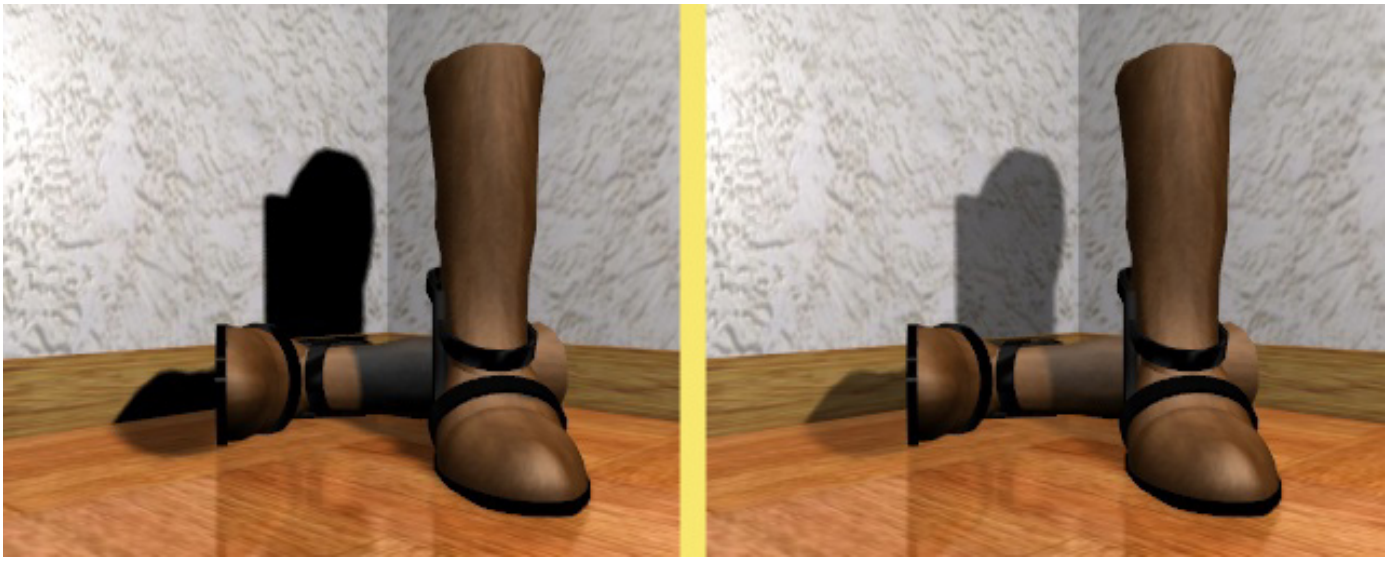
BufSi 2560, SpotSi 60.00 and Samples 5; It's up to you to if you really need such quality!

Fine tuning your shadows

"Only Shadow" button and "Energy" value

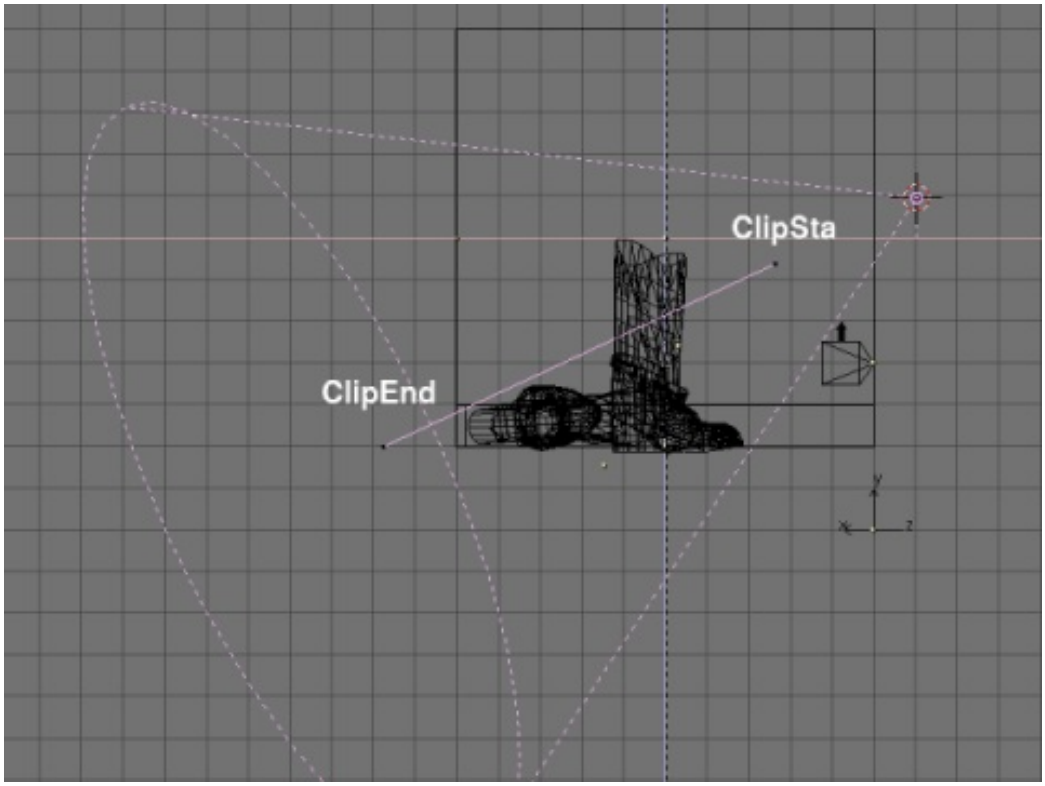
So far, we used our Spot to cast both light and shadow. This is why in our latest renderings, we can see the spot light on the floor and on the walls. Most of the time, this isn't the effect you want to achieve. Instead, you would like to dissociate shadows from the light source. This is easily done : set a few Lamps, Hemis, or Suns (whatever you like) for your whole scene. Then add many Spot lamps (ideally, one for each object for which you'd like to see a shadow, or at least one for a group of close casting shadows objects) for which you turn the "Only shadow" button along with the "Shadows" button. Et voilà! Your Spots only cast shadows and don't add to the luminosity of the scene!

Alas, the shadow deepness increases dramatically. In order to compensate this, you can play with the Energy value of the Spot. As a rule of thumb, a middle value gives almost the same result as a Spot casting both light and shadows. But sometimes, you'll like it with pitch dark shadows!



Lamp, "Only shadow" Spot with Energy at 1.0; The same, but with Spot Energy at 0.5

"ClipSta" and "ClipEnd&qu



These two parameters set how light and shadows are dealt with in the Spot vicinity. They are visualized by a pink segment running along the Spot direction, such as the example above.

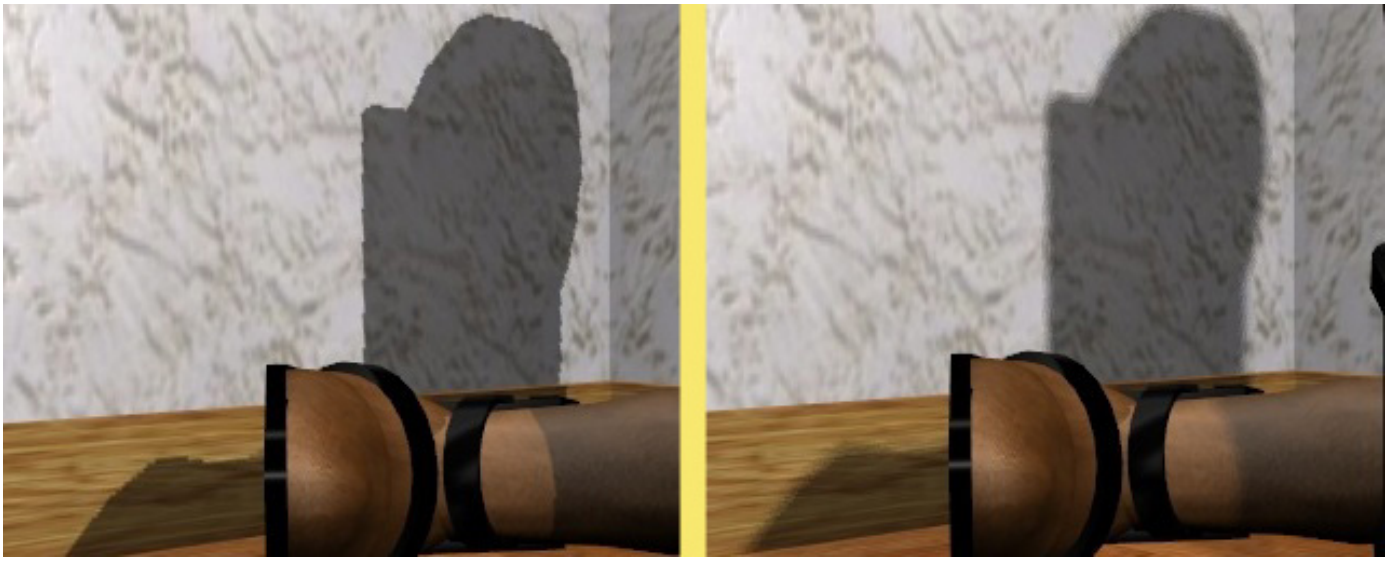
In fact, every face of any mesh closer to the Spot than the ClipSta value always has light, and every face of any mesh beyond the ClipEnd value always has shadows. Any face of any mesh between these two current values has its pixels given a light or shadow value according to the data stored into the shadow buffer.

There are two rules that you should follow in order to always get the neater shadows:

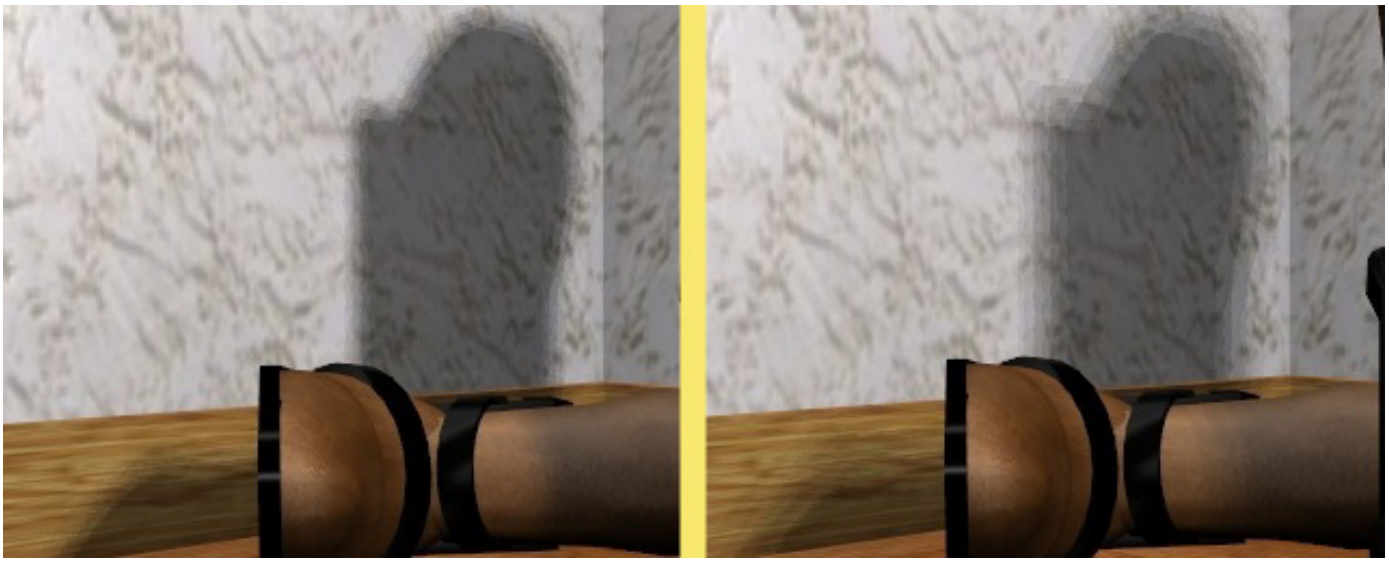
- Set the range between ClipSta and ClipEnd the smallest possible.
- Set ClipSta the highest possible.

Soft

This defines the size of the area being sampled (see Samples, before). The higher the value, the smoother the borders of the shadow. In all the previous renderings, Soft was set to 3.00, but here you wan see what happens to the pictures when set to higher or lower values.



Left: BufSi 512, Soft 1.00; Right: BufSi 512, Soft 5.00



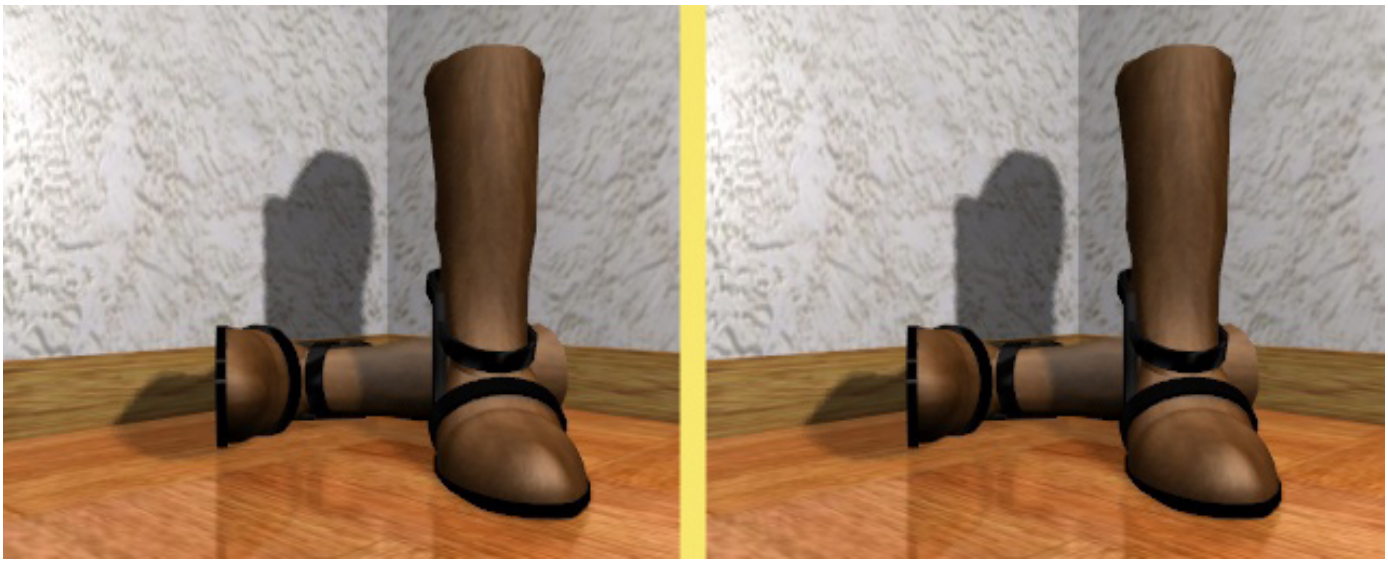
Left: BufSi 512, Soft 10.00; Right: BufSi 512, Soft 20.00

Using a low Soft value like 3.00 is perfect most of the time, and it should be used as default value whenever appropriate. Higher values soften the borders to an extent that remind lights emitted by large surfaces (just like white neons in supermarkets), with according fuzzy shadows.

Bias

Sometimes strange patterns appear in the shadow area, when Bias is set to a low value and when the range between ClipSta and ClipEnd is low (as is usually recommended). To prevent this, you should always set Bias to the highest value possible.

Left: Bias set to 0.01, Right: Bias set to 1.00



Left: Bias set to 3.00; Right: Bias set to 5.00

The patterns are clearly visible when Bias is set to 0.01 and are progressively smoothened when this parameter is increased. It's hard to tell at all if there's a pattern in the rendering with the Bias set to 3.00, but the picture seems to be a little darker than the one with Bias set to 5.00. Confused? Nevermind! Set Bias to 5.00 without even thinking about it!

Final conclusion

You are now deeply familiar with casting shadows within Blender. Given time, you will play with light and shadow concepts without even thinking about it, and enhance the quality and realism of your pics by 100%. But before that, keep it simple and straight-forward:

- Try to set all your SpotSi within 45.00 and 60.00 degrees values, avoid setting them above 90 degrees
- Set Samples to 5
- Try to set BufSi to 768 or 1024
- Set the range between ClipSta and ClipEnd the smallest possible
- Set ClipSta the highest possible
- Set Soft to 3.00
- Set Bias to 5.00

Happy blending!

Feedback

dreamscape	2001 03 01
Hey, Thanks alot. Before i was just a complete idiot when it came to shadow effects!	
BarryBonds	2001 01 29
Thanks a lot!	
iMyst	2001 01 28
(wolf-whistle) shaweet! -iMyst	
Jamesk	2001 01 17
An enlightning tute, perfectly spotless...	
Cujo31	2001 01 17
Great Job! Excellent tutorial :)	
Zycho	2001 01 15
You can find more of Olivers tutes and this one on http://www.blender-cafe.org/ Enjoy!	
James15185	2001 01 15
This is a great tutor, I hope that Olivier Saraja puts out more of these. It is well thought out and easy to understand. Thanks.	
Tiddles	2001 01 14
another great Tut.... Thnx u all!	
ThreeDee	2001 01 14
Well, stupidly I thought I knew all about spots. I was wrong! Thank you! :D	
lemming	2001 01 14
Ahhh brilliant - yet again !!!	
java2	2001 01 13
Thnx to people who take their time to help beginners like me. And thnx to blender programmers...i love this stuff.	
stephen2002	2001 01 13
very good, this should clear up a lot of confusion about shadows	
Toonami_fan	2001 01 13
NIce one! <MATT>	

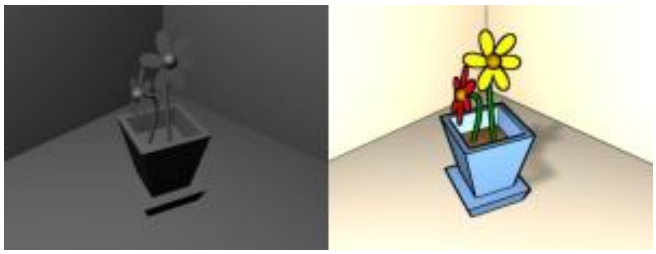
www.blender.nl
this document is online at http://www.blender.nl/showitem?id=80

Tutorials
2000 11 27
id80

Toon Shading

Chris Garrett

This tutorial presents the techniques that I use to achieve a toon style render directly in Blender without any post production in another program. In this tutorial we will modify a basic Blender scene of a simple flower pot to look toon shaded by changing the lighting and materials. At the end the left image should look like the right image.



Begin (left), end (right)

This tutorial is designed for users who have a reasonable knowledge of the use of Blender. Before you start this tutorial you should know how to add lamps, move and rotate objects and add materials to the surface of objects. If you have trouble with these concepts visit the Blender web site and examine some of the getting started tutorials first.

Before we can start you will need the flower pot model - you can download it below.

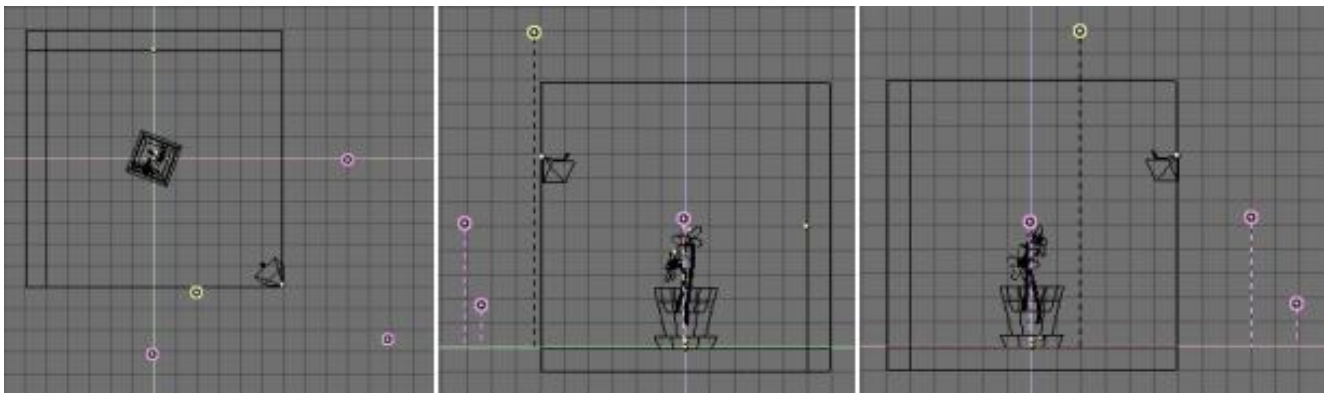
Once you have downloaded and unzipped the model open it in Blender.

Download: toon_shade_tutor.zip

Step 1: Lighting

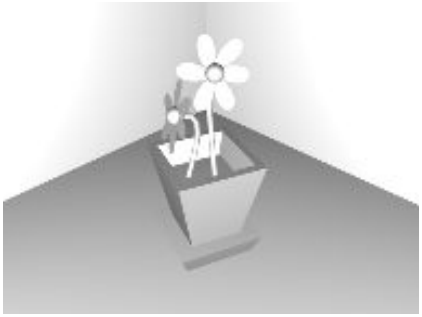
The first thing we need to do to the scene is to brighten it up. When lighting a scene I use a minimum of four lights, three fill lights and one detail light. The fill lights are placed around the camera, one on each side and one behind. The detail light is a spot light placed above the scene pointing at the most important thing in the scene, it is used to highlight the subject and to supply shadows.

The lights in toon shading are particularly important, there must be enough light or brightness on the faces to supply contrast to the edges. If the edges are not visible it won't look as good.



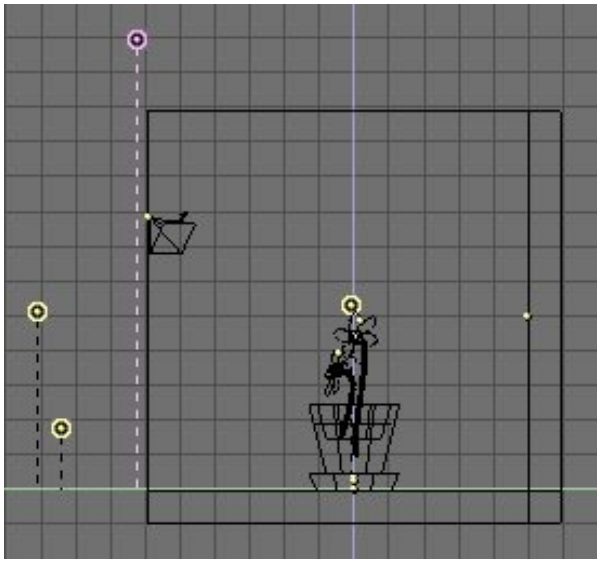
Lamps (top/side/front)

As shown here the extra lamps have added the brightness that we need, all the edges of the model are clearly visible and no face is completely black.



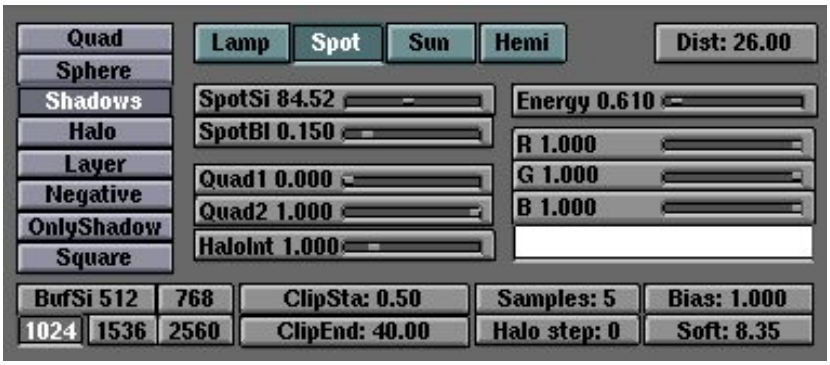
We now need to modify the original light to cast shadows and to point at the flower pot.

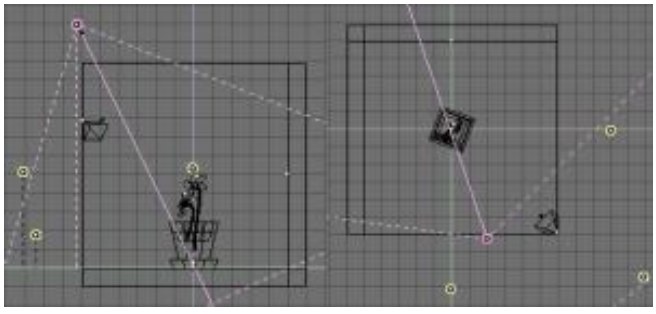
Select the original light.



Press the light button or press F4.

In the buttons window set the following properties for the original light.

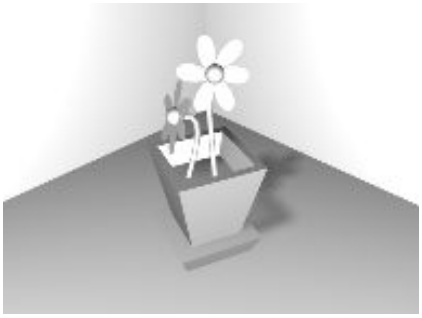




(side/top)

Rotate the light so that it points at the center of the flower pot.

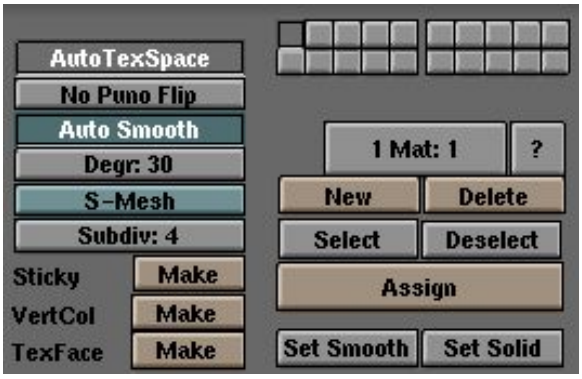
The lighting is now complete, the modifications to the original light have added the highlights and shadow that were needed.



Step 2: the object rendering material

I found that the toon shading looks better if objects are set to render smooth rather than solid. Setting them to render smooth appears to make the edges more defined.

Select each object in turn and click the set smooth button in the edit buttons window (F9).



Setting materials to auto smooth.

Step 3: Adding Materials to objects

Now we get to the fun bit, materials. Materials in toon shading need to be flat with little or no specularity. Generally when creating materials for toon shading the colors need to contrast. Try not to create dark colors or overlap colors of similar style. The colors need to contrast so that the edges are clearly visible.

Select the flower pot model.

Floor (left) and Walls (right) materials

Flower stem (left) and Lower Petal (right) materials

Upper Petals (left) and Flower Center (right) materials

Pot Dirt material

Once you have added all the materials to the objects you should have a scene that looks similar to the following. It is then time to move onto the last step, the edges.

Step 4: Edges

With Ton's great addition to 1.76, the edge control, adding cartoon edges to an object is as simple as clicking a button.

Using the Display Buttons (F10), click in the Edge button and set the Eint: value to 150.

The higher you set the Eint: value the more edges will be outlined. In a simple scene like this setting the value very high is not a problem, but in more complex scenes be careful how high you set it. If the value is very high and the objects have many faces, the faces themselves will start to get outlines. The goal that we are seeking is to get black lines only on the edges of the object not the internal faces as well.

Experiment with the Eint: value to get the look that you require. Sometimes a scene with only a few edges highlighted can look much better than a scene where every edge has a black line.

If everything has gone well you should now have a image that looks like this!

Feedback

GammaRayQ21	2000 11 27
This tutorial will help me very much.	
zoo	2000 11 27
Great, I see the tutorials entering the site! And Blender can not have enough tutorials. Thanx Chris.	

www.blender.nl
this document is online at http://www.blender.nl/showitem.php?id=20

Tutorials
2000 08 02

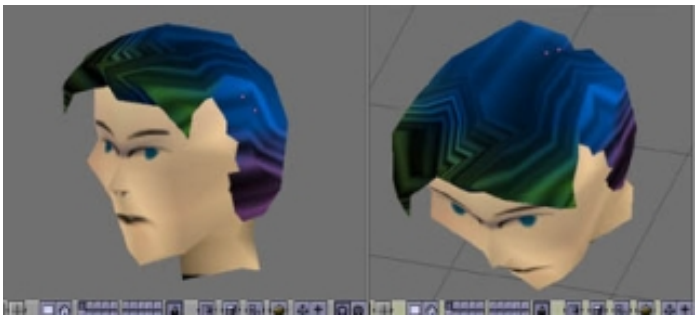
UV Mapping

Samo Korosec

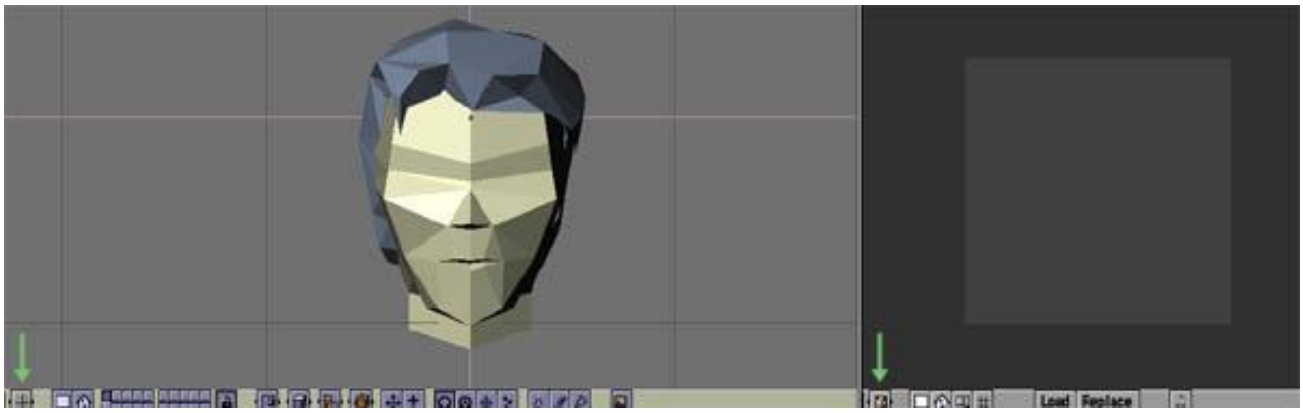
id20

Since Blender 2.0 - 'gameBlender' - was just around the corner, NaN added the UV texturing tool to the blender 1.x series to enable users "prepare" themselves for game content creation, by practicing UV texturing along with low-polygon count modeling. This tutorial was written for 1.8, but everything still works the same in 2.0.

Well, there might have been some confusion about the UV texturing, since Ton suggested to me today, that I should write a UV texturing tutorial :o) Here it is!



Final result.



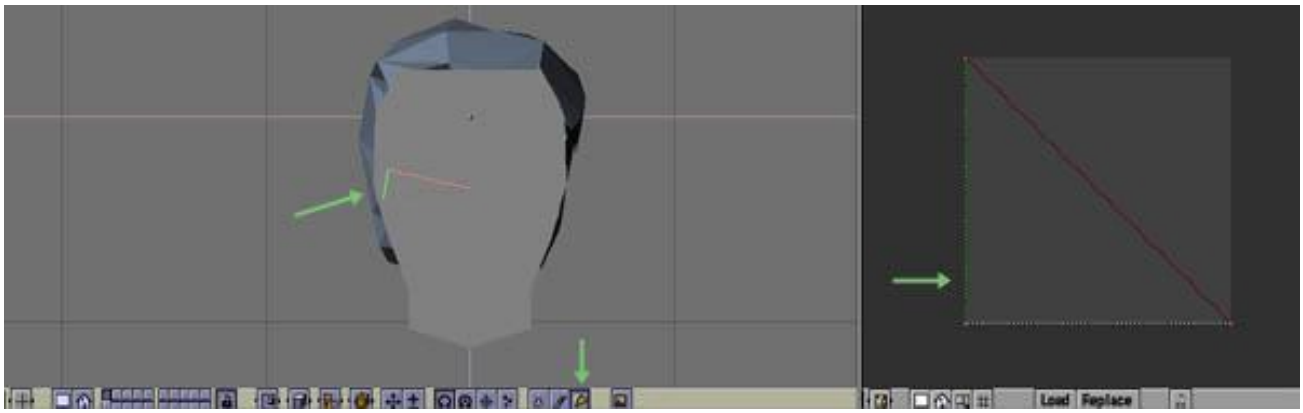
We start with the finished object and the image window opened.

You will start off with a file you created your model in (be it a face, character, vehicle or whatever). Once loaded, you will need two windows to work efficiently - the 3D window (Shift F5) and the image window (Shift F10).

The 3D window will represent your model, while the image window will show your textures and allow you manipulating the UV coordinates. The green arrows indicate the buttons that tell you the type of the window, the left being your 3D and the right your image window.

You can download an example .blend file fot the face here.

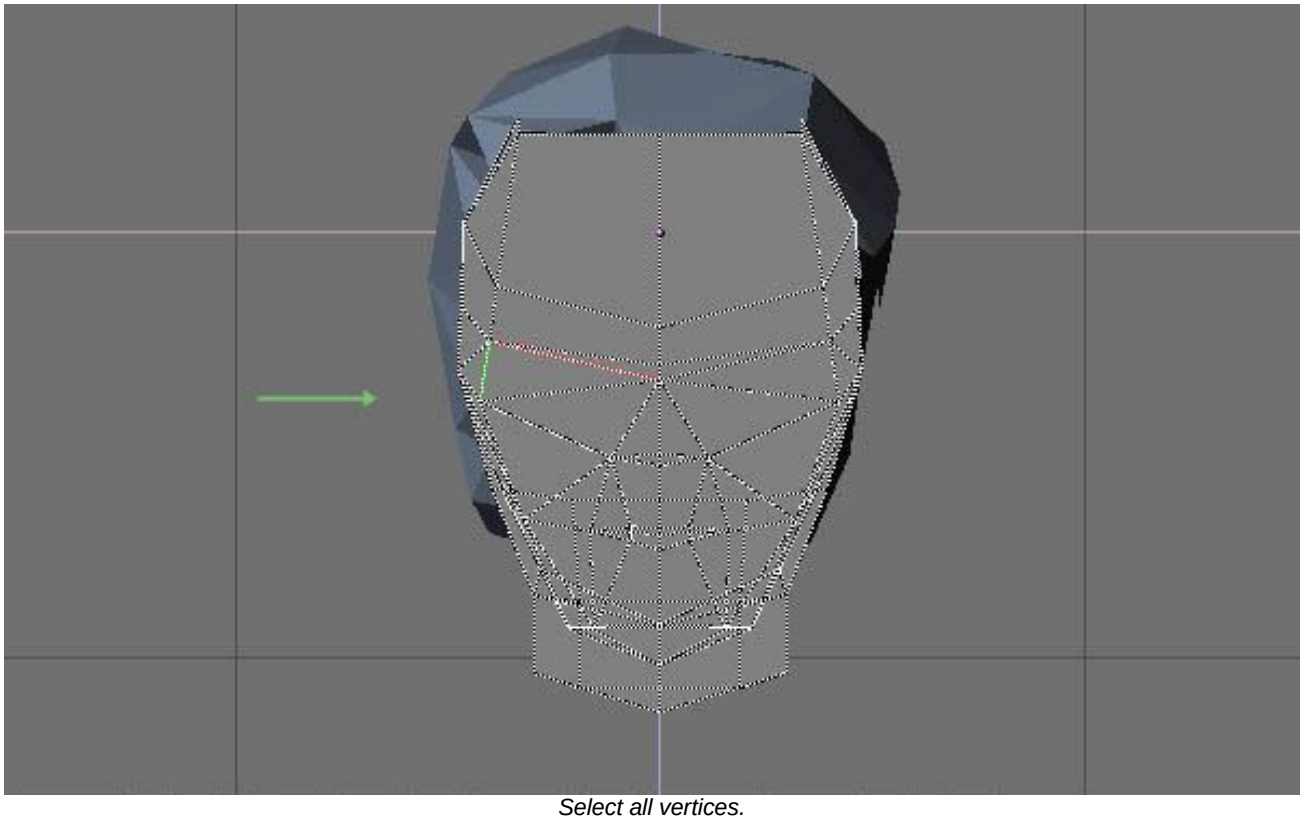
Download: uvtextstart.blend



With your object selected go into face select mode.

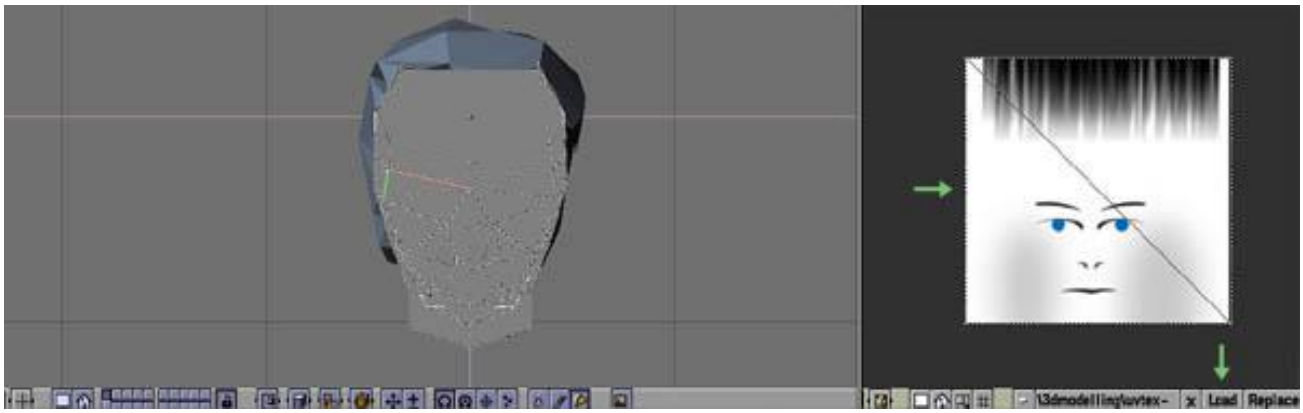
The key to UV texturing is the face select mode (F), where you tell Blender which faces' UV coordinates to use. The arrows show the current active (and selected) face - left, the icon for face-select mode - middle, and the representation of the face in your image space - right.

You can select faces with the right mouse button. Holding down shift while clicking with the mouse allows you to add more faces to your selection or deselect some, without affecting the state of other faces, that not accessed with the mouse. You can basically use the same selection techniques as you do in your everyday Blender work.



Select all vertices.

However we do want to map our whole face model and instead of selecting each face, we select all by pressing A. However, a face must be selected before that anyway, to appear "active" (see the green arrow), as some other face attributes must have a origin - the active face.



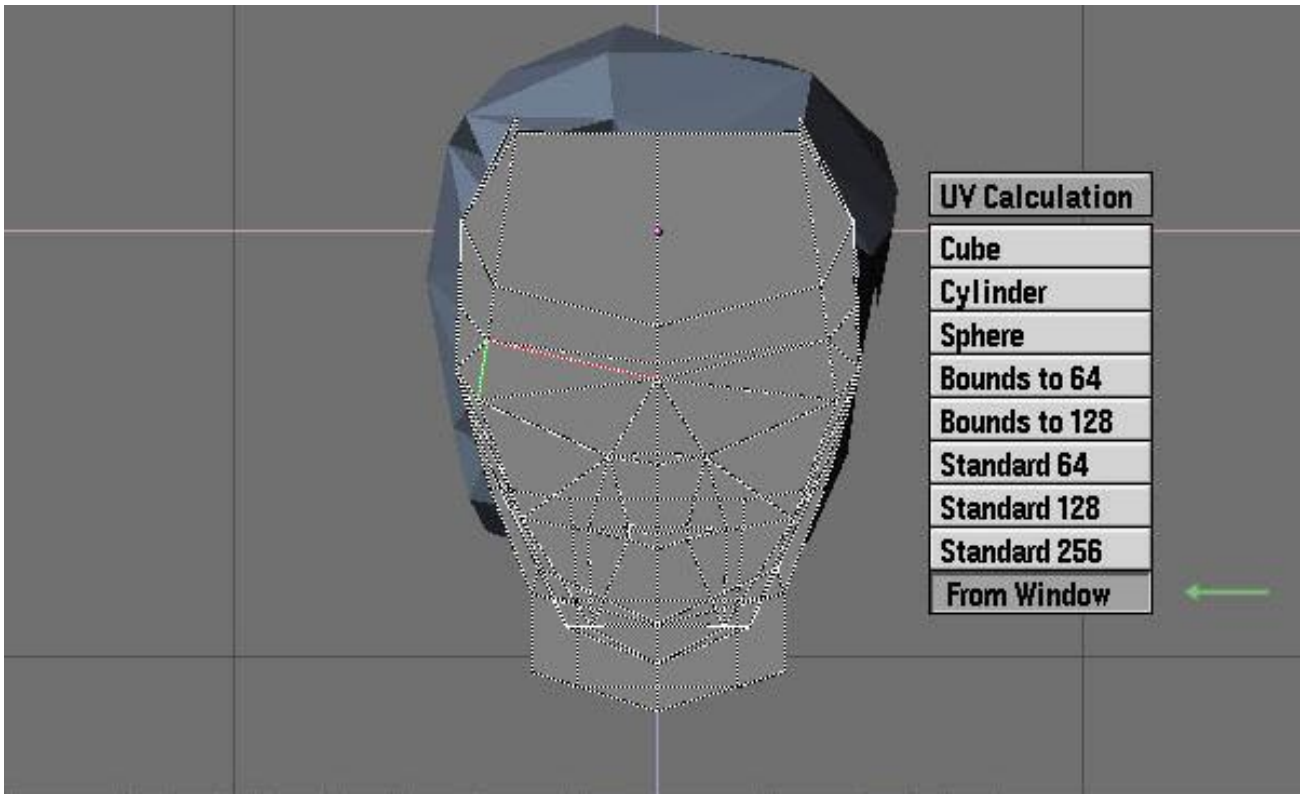
Load an image in the image window.

The next step you need to do is to actually load the texture (the green arrows indicate the '**load**' button and the area where the texture should appear), where it's width and height have to be a power of two and also have the same dimension. Valid examples would be image sizes like: 8x8, 16x16, 32x32, 64x64, 128x128, 256x256, 512x512,...

This is due to the way OpenGL handles textures; you could use a texture with its height and width not being the same, however most OpenGL hardware (especially games adapters) will then stretch the texture to fit in their memory.

(You can download the texture here:)

Download: uv_map.jpg



Map UV coordinates.

Once you have your faces selected, you need to load their UV coordinates into the image window. To do so, you need to press U and choose one of the options presented to you. I chose "From Window" - that used the 3d window to map the UV coordinates - hence, the image window contained a "front" view of the objects' coordinates.

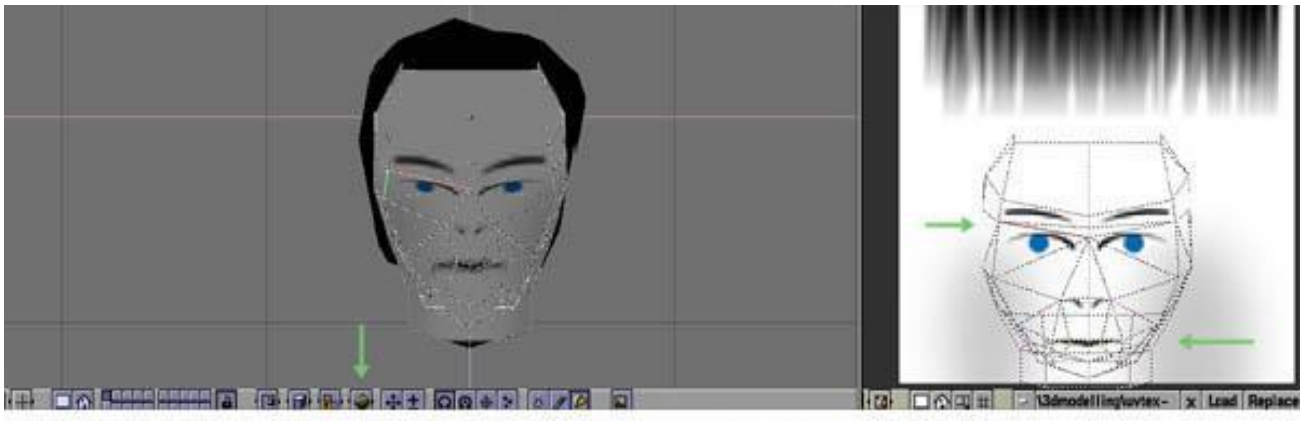
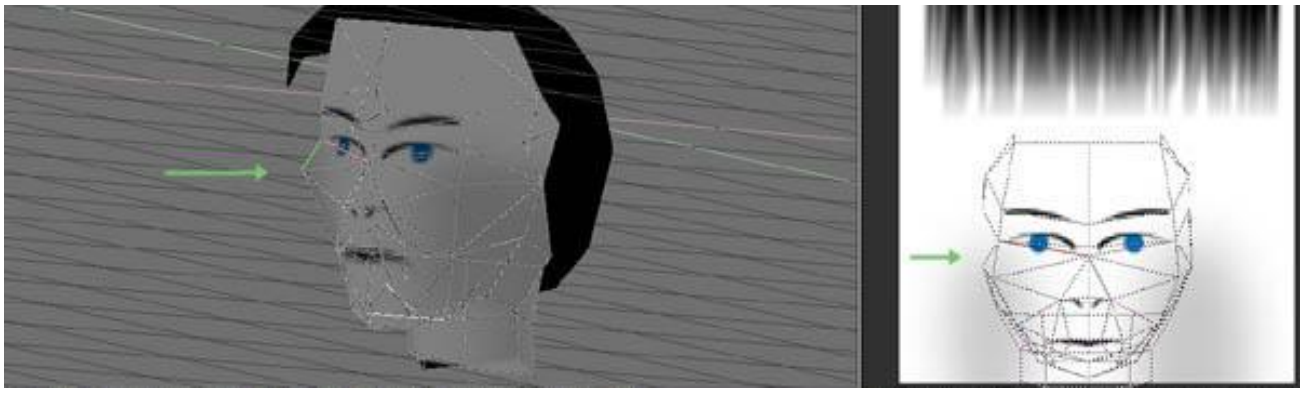


FIG. 6 - YOU CAN NOW MOVE THE VERTICES IN THE IMAGE WINDOW TO ADJUST MAPPING

You can now move the vertices in the image window to adjust mapping.

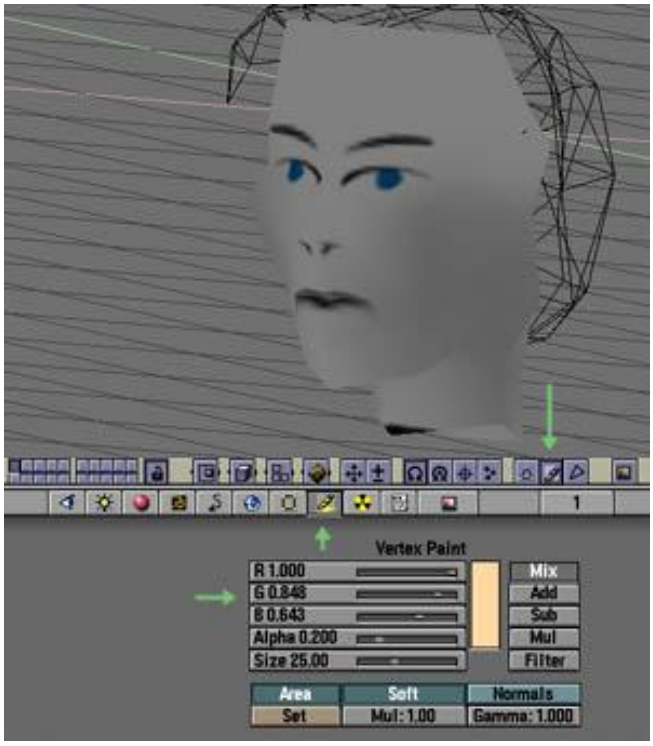
You need to turn on the textured view (left green arrow) now, in order to see the adjustments of the UV coordinates you perform (right arrows). This should give you the first UV texturing results.

Editing the UV coordinates works the same way as editing normal vertices in edit mode. You can play around by selecting a group of the UV coordinates and scaling, rotating or moving them.



Fine tuning your mapping.

However, you might soon find yourself in a situation, where you are not sure what impact your UV editing actions have - feel free to rotate the 3D view, zoom it in and out etc. You can also move and zoom the image windows, allowing very detailed mapping work.



Go to vertex paint mode.

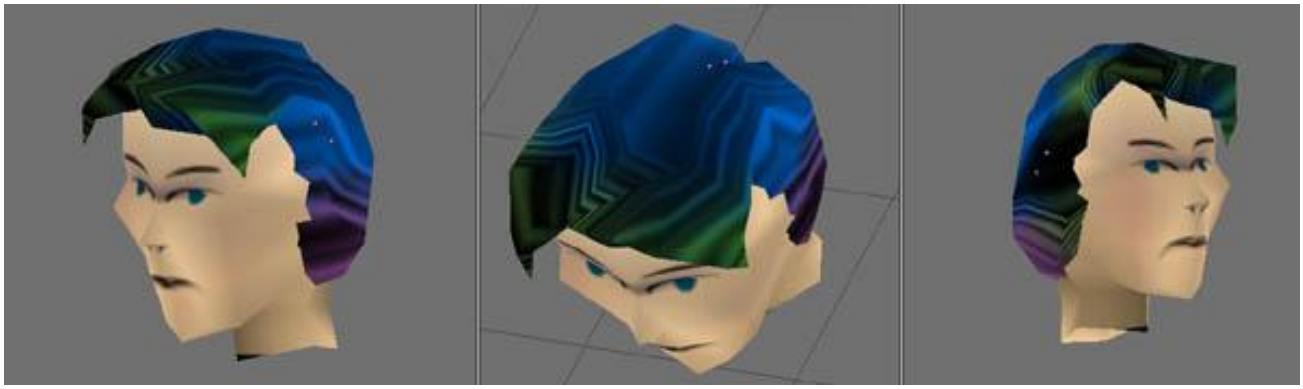
Once we have out textures in place, our friend still looks kind of sick - lain gray is no color for a face!

Exit face select mode by pressing f and enter vertex paint mode by pressing V or clicking the button indicated by the right arrow. The other arrows indicate the vertex paint buttons button (funny, eh?) and the sliders to choose the right colors.

While in vertex-paint mode, your left mouse button lets you paint on object, while your right mouse button (thanks, Rob!) lets you choose colors beneath the mouse cursor. I tried to get a skin-like color and just painted away.

Vertex painted mesh with UV texturing

After finishing with vertex painting, we can enjoy our new-textured head. It looks very clean and nice with black hair, however I added a "hair" pattern sort of on top of my face texture and you can use it..



Repeat the same for the hair and enjoy the result.

..to make a punk! :o)

I hope this little tutorial was helpful.