BLENDER BASICS

A Beginner's Visual Guide to 3D Modelling

Alistair Stewart

Digital Skills

Blender Basics

A Beginner's Visual Guide to 3D Modelling

Alistair Stewart

Digital Skills
Milton
Barr
Ayrshire
KA26 9TY
UK

www.Digital-Skills.co.uk

www.digital-skills.co.uk
tel: +44(0)1465 861 638

Blender was created by Ton Roosendaal.

Book Started Publishing: January 2024

Title: Blender Basics
ISBN: to be assigned

Other Titles Available:

Hands On AppGameKit Studio Volume 1
Hands On AppGameKit Studio Volume 2
Hands On C++17

Introduction

3D Modelling

Using a 3D modelling package allows us to create static objects or scenes which can be:

printed using a 3D printer
imported into a video game
used to create a photo-realistic image
form the landscape for a Virtual Reality scenario

The modelling package also allows us to create animations and these can be used to:

create an animated video
embedded into real world video to create special effects
export to a video game where the game can control execution of the animation

Blender 4.1 is, at the time of writing, the latest incarnation of this powerful, professional-level 3D modelling package. And, despite the fact that other, similar packages are extremely expensive, Blender is free! You can download **Blender 3.1** from *www.blender.org*.

This Book

This book is aimed at the absolute beginner in 3D modelling. It will guide you, using a graphics-based, approach, through the fundamentals of 3D modelling and the specifics of using Blender 4.1. The pages are written in a way that allows you to follow along, creating your own models as you read through the pages, so that rather than your learning being a passive experience, it is a very active one which will not only make working through the book more fun, but will also help to develop and retain the skill set of a 3D modeller.

The Videos

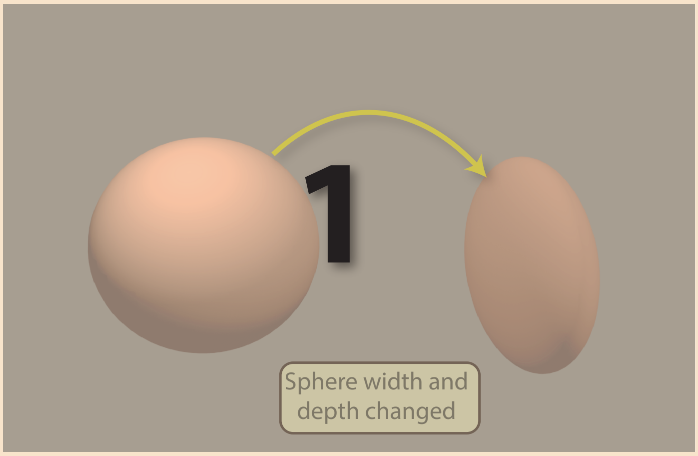
We have also developed a set of videos which are designed to be used in conjunction with the book. The videos are available on our YouTube channel. You'll also find a link to the relevant video in each section of the book - look for the symbol. Click on the symbol to watch the video.

How to use the Package

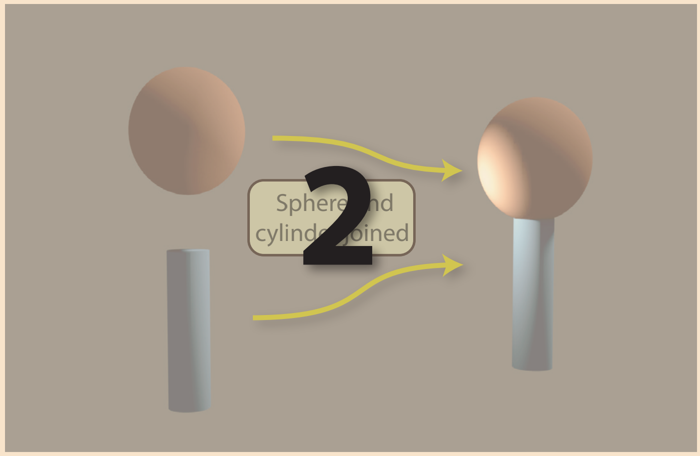
A suggested approach to making use of this video/book combination is to watch the video for a particular section (most are less than 10 minutes) and then use the corresponding book section as a reminder of the salient points covered by the video. Once you have enough basic knowledge of modelling, we will also suggest activities to try for yourself which will reinforce the learning process and build up your modelling skills.

Most pages in this book are designed as a set of six panels which should be read from left to right, top to bottom:

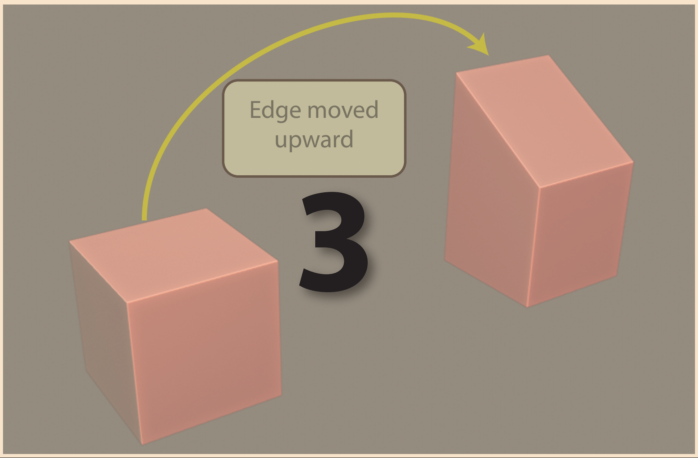
When creating models there are four basic approaches that may be used separately or in conjunction. The first of these is to modify the dimensions of a primitive.



A second option is to join two or more primitives to create a new shape.

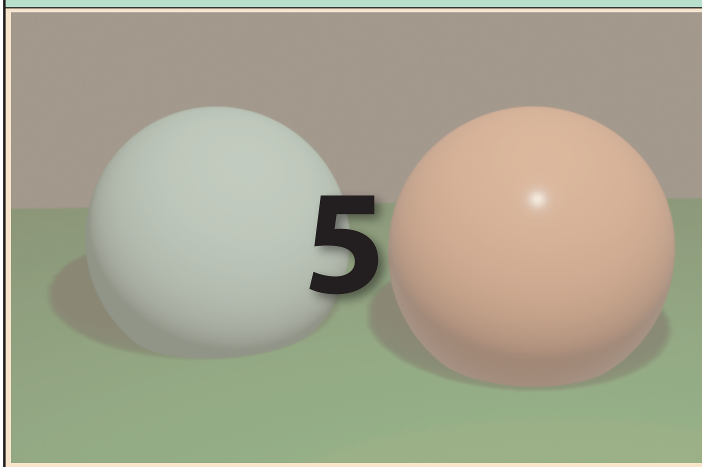


The third, and most used option when modelling man-made, rigid shapes (known as **hard body** objects) is to reposition the vertices, edges and/or faces that make up a primitive.

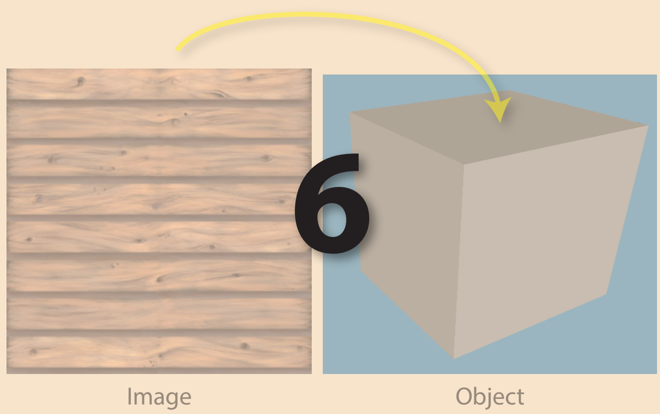


Finally, when creating organic shapes such as people or animals many modellers use sculpting. This technique is somewhat similar to molding clay and requires a fair degree of artistic talent!

Adding a **material** to an object allows us to define that object's final colour and reflectivity. For example, a material can give an object a dull or shiny appearance or have it appear transparent.

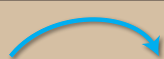


Once an object has been assigned a material it can also be given a **texture**. A texture normally consists of one or more images which are assigned to the object.



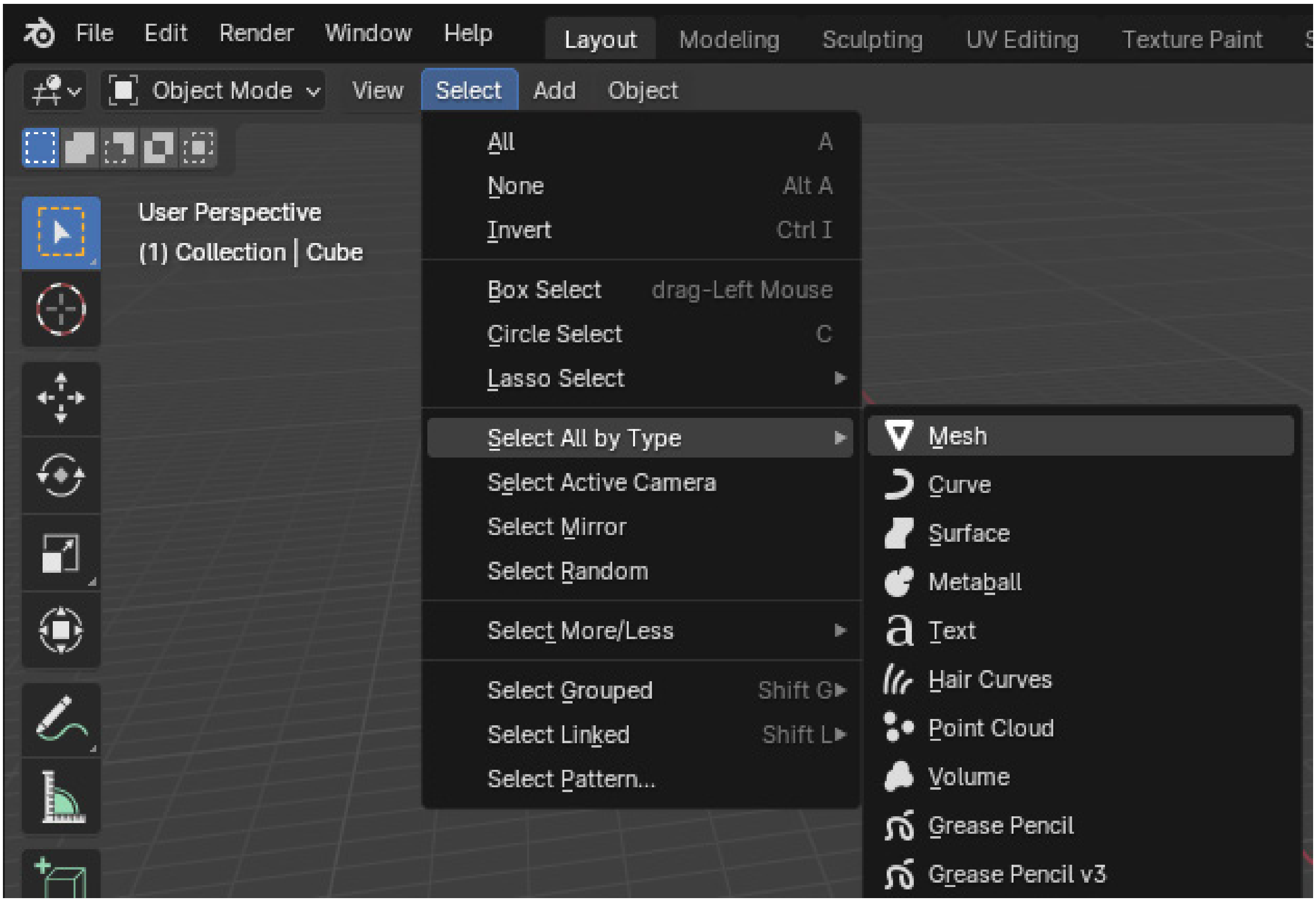
Book Conventions

Symbols are used to represent mouse and keyboard operations:

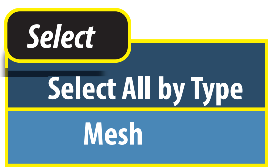
Symbol	Meaning
	Press left mouse button
	Release left mouse button
	Press right mouse button
	Release right mouse button
	Press middle mouse button
	Release middle mouse button
	Double click left mouse button
	Scroll mouse wheel up/down button
	Scroll mouse wheel up button
	Scroll mouse wheel down button
	Drag mouse, left button down
	Drag mouse, right button down
	Drag mouse, middle button down
	Move mouse, no buttons down
	Path of mouse drag operation
	Perform mouse operation at specified position (tip of arrow)
	Press specified key on keyboard
	Enter a value from keyboard
???? + ?	Hold down specified key while pressing second key
???? + ???? + ?	Hold down specified keys while pressing third key
? → ?	Press first key then second

Various Blender menus and attribute settings are represented in a shortened form after their first occurence in the book as shown below:

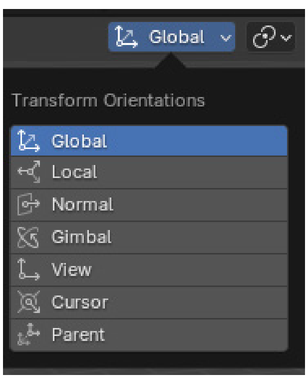
Menu entry (in Blender):



Menu Entry (in book):



Dropdown Field (in Blender):



Dropdown Field (in book):

BACKGROUND

The commonest approach to 3D modelling is to start with a basic 3D shape, then modify it to create the final shape. Next, we add one or more materials to our shape to give it colour, shine and transparency as required. A texture image can be assigned to a material to give the shape the appearance of being constructed from some medium such as wood, stone or brick.   Finally, our scene is processed by a render engine which creates a final, realistic image.

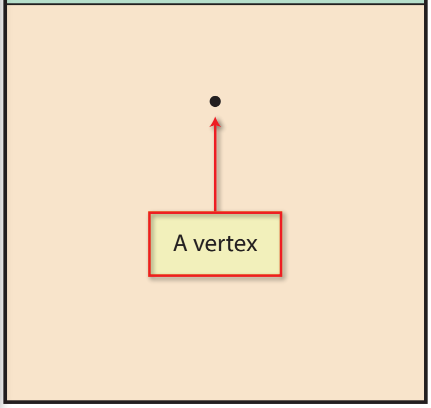
Of course, this is a greatly simplified view of the stages required and where we want to create an animated sequence rather than a still image there are many more steps required in the process.

In this section we'll have a brief look at the main concepts that are involved in producing static 3D models.

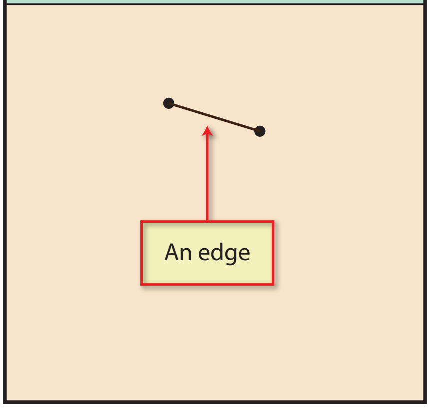
This background knowledge will give us a good foundation for the remaining sections of the book where we will learn in detail how to create our models using Blender - a free 3D modelling package.

3D Modelling Basics

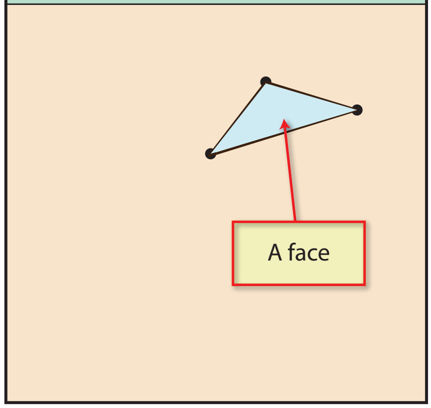
The fundamental building block of most 3D models is the **vertex** – a single point in space.



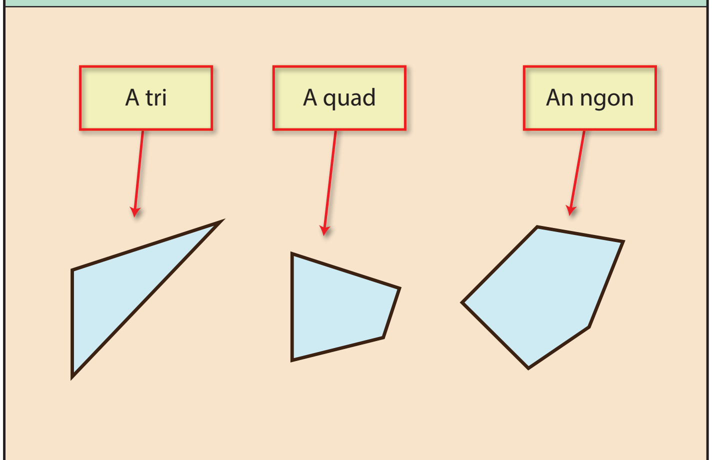
Two vertices can be joined together by an **edge**.



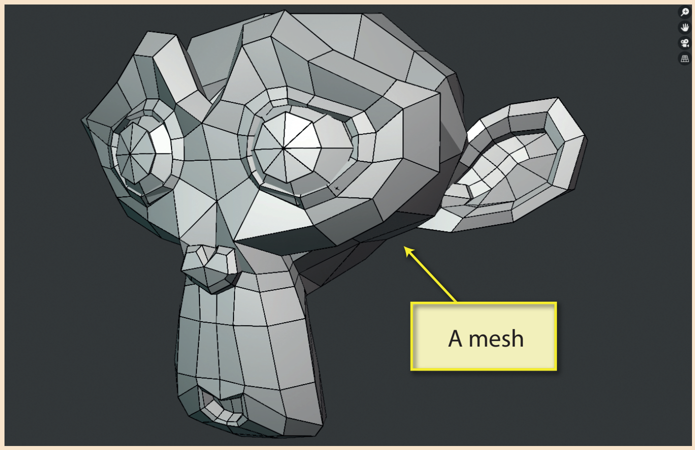
Three or more edges can be joined to create a **face**.



In Blender, a face with three edges is known as a **tri** (as in triangle); one with four edges is called a **quad** (as in quadrilateral); and a face with more than four edges is termed an **ngon** (short for *n*-sided polygon and sometimes written as ***n-gon***).

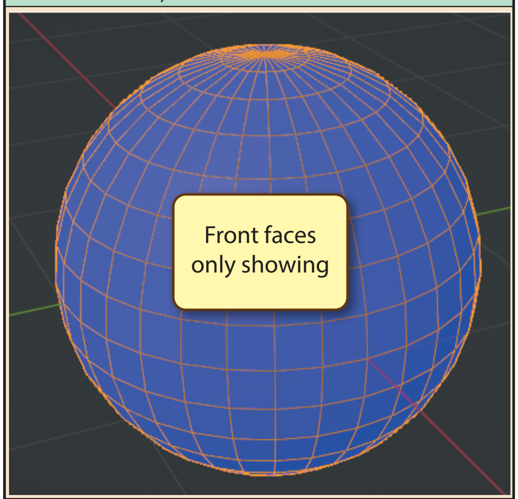


By combining a set of faces into a group (known as a **mesh**), we can create a 3D shape. Note that a vertex may be shared by more than one edge and that an edge may be shared by more than one face.

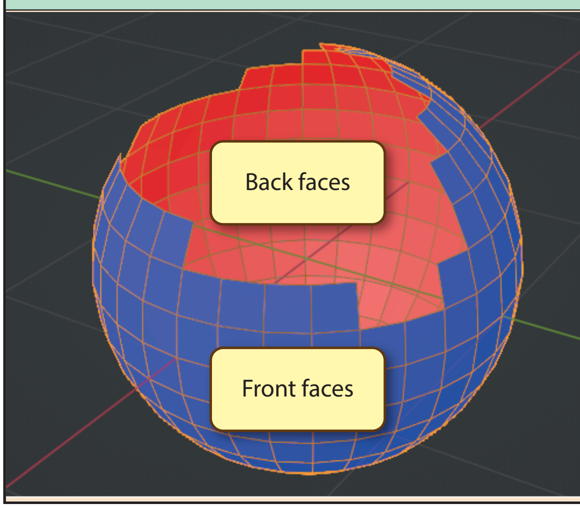


Like a coin, every face has two sides: one side is designed to be seen by the viewer (known as the **front face** or just **face**) and the other side hidden (the **back face**).

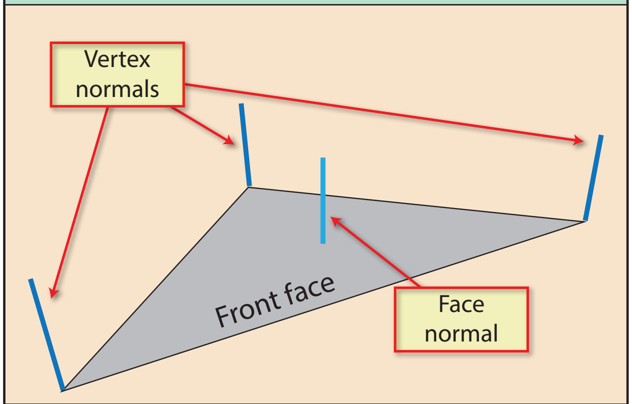
If we have a sphere-like mesh, then all we can see are the **front faces** of the object (shown below in blue).



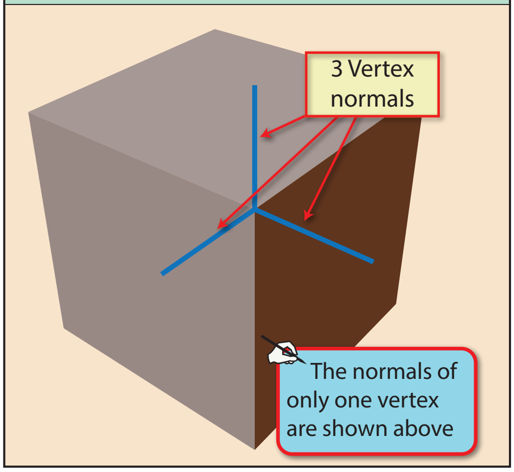
However, if we remove part of the sphere, some **back faces** are exposed (shown in red).



Another feature of every face is a set of **normals**. Normals may be thought of as invisible lines radiating from the front face (**face normals**) and every vertex (**vertex normals**). Face normals are always perpendicular to their associated face and vertex normals are perpendicular to the two edges that meet there.

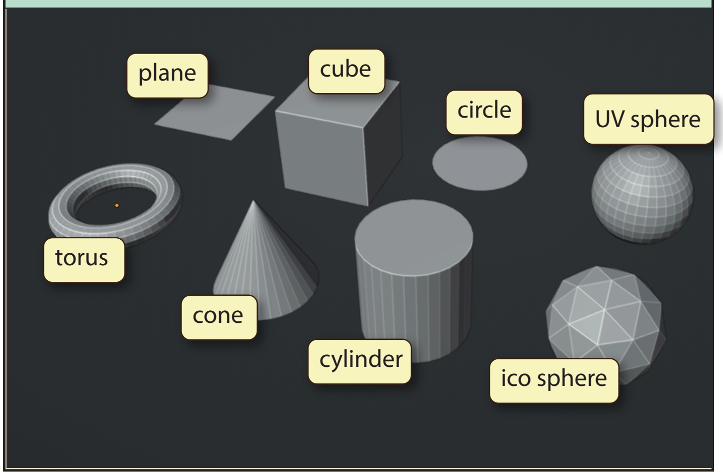


Where more than one face shares a vertex, a vertex normal is created for each pair of edges.

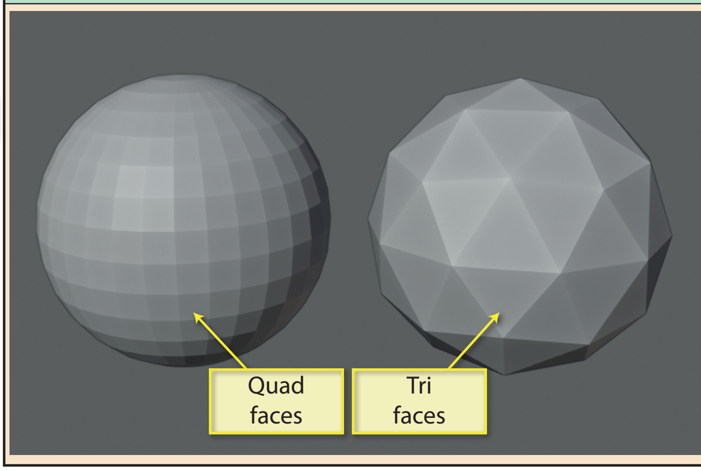


Blender uses normals, not only to determine which side of a face is the front face, but also to calculate shadows and reflections.

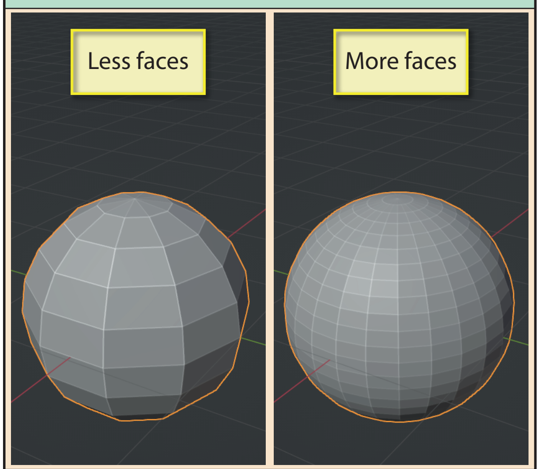
Generally, models are created by joining and/or modifying meshes. Most modelling packages offer a set of basic shapes which can be created automatically. These are known as **mesh primitives**. The commonest primitives offered by Blender are shown below.



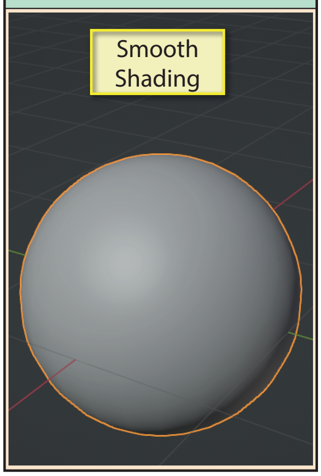
Notice that the ico sphere has **tri** faces while the UV sphere has (mostly) **quad** faces. ***Quads*** are considered the most desirable shape for faces when modelling.



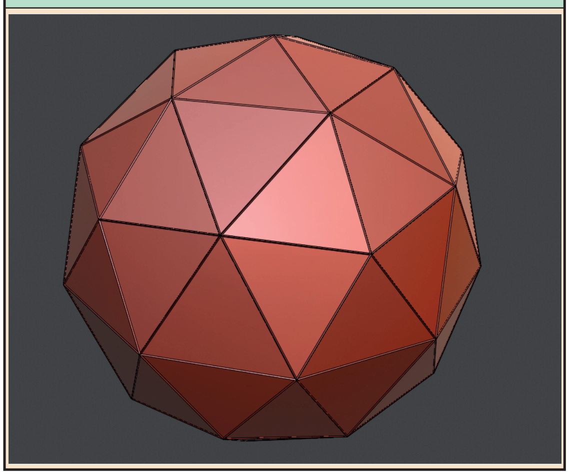
For curved shapes, the more faces a mesh contains the more realistic the model will appear. The number of polygons in a model is usually referred to as the **polycount**.



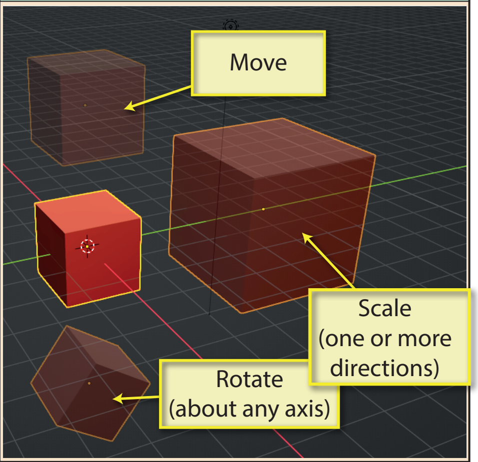
An even more effective way of creating the illusion of a curved surface is to adjust the shading.



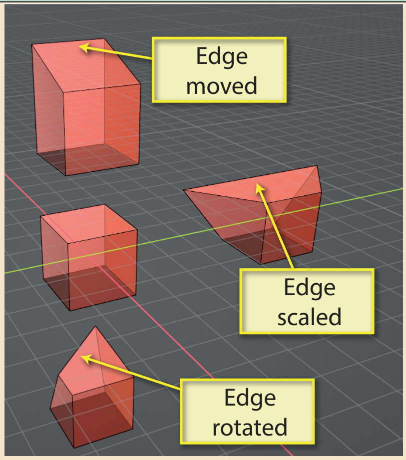
While modelling, we can assign a temporary colour to the objects in our scene. This colour has no effect on the final result but may help make objects easier to see during the modelling process.



When creating our scene we can move, rotate or scale a complete mesh...

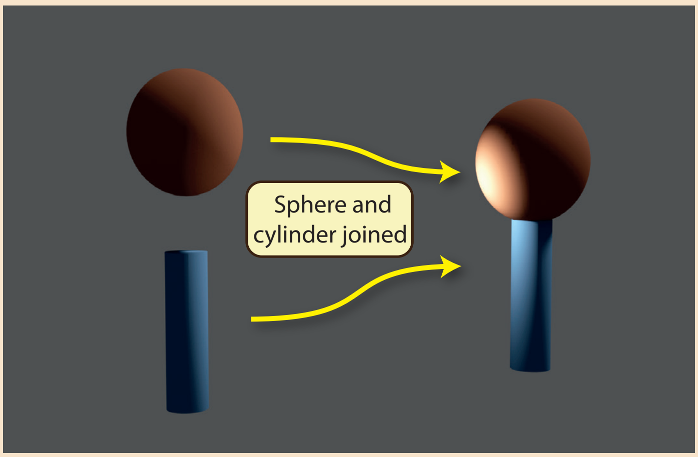


... or we can move, rotate or scale parts of the mesh.

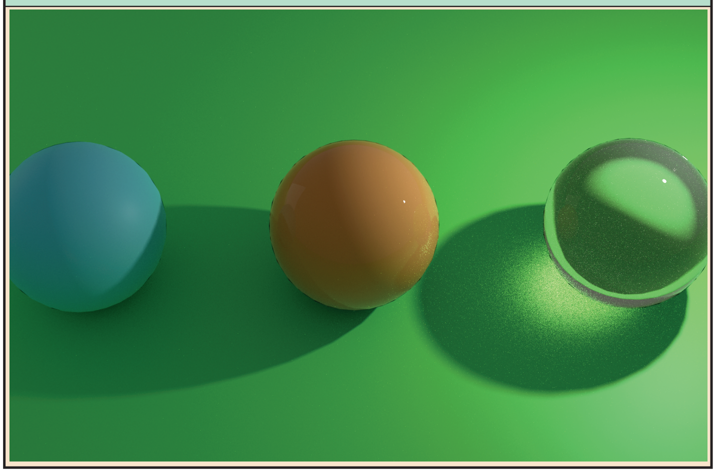


When creating organic shapes such as people or animals many modellers use sculpting. This technique is somewhat similar to molding clay and requires a fair degree of artistic talent!

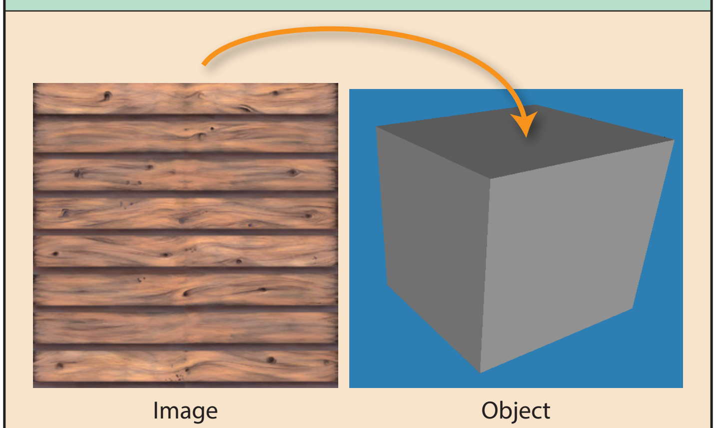
Another option is to join two or more primitives to create a new shape.



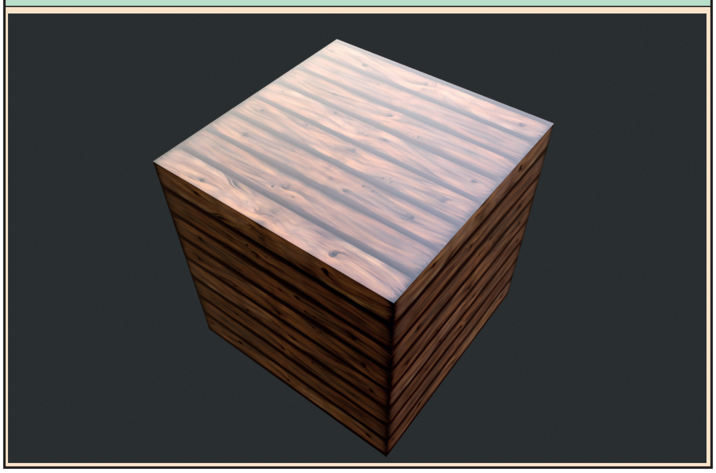
Adding a **material** to an object allows us to define that object's final characteristics such as colour, reflectivity and transparency. For example, a material can give an object a dull or shiny finish or have it appear to be made of glass.



Once an object has been assigned a material it can also be given a **texture**. A texture normally consists of one or more images.

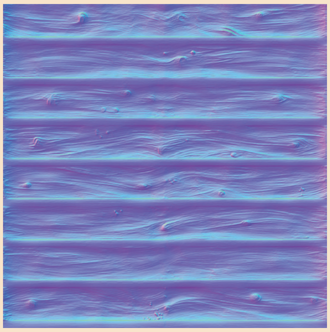


This adds a texture to the object giving the impression that the object is constructed from the material shown in the image.



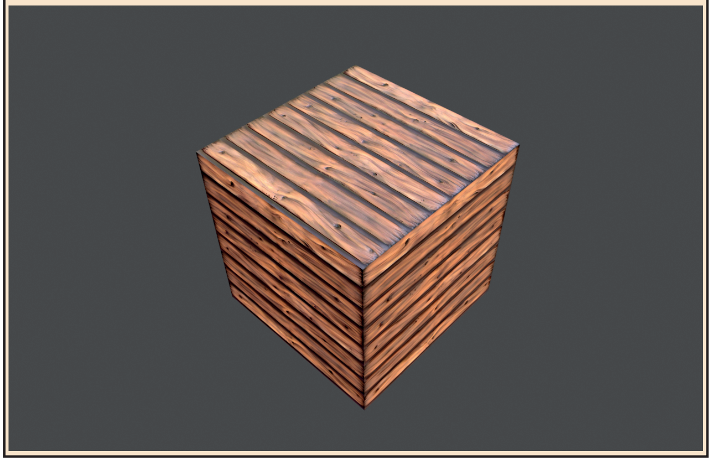
To add further realism to an object we can assign a **normal map image** to the object...

Texture Image

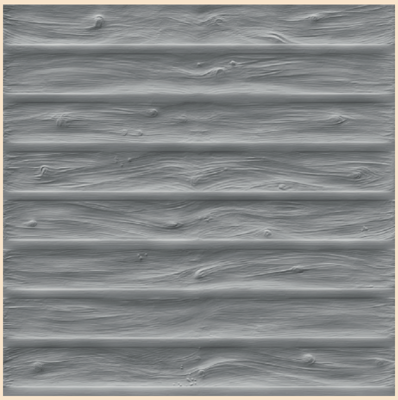


Normal Map Image

By adding the normal map image, the object appears to have a rougher, pitted surface (depending on the nature of the new image) without actually adding more faces to the object itself.

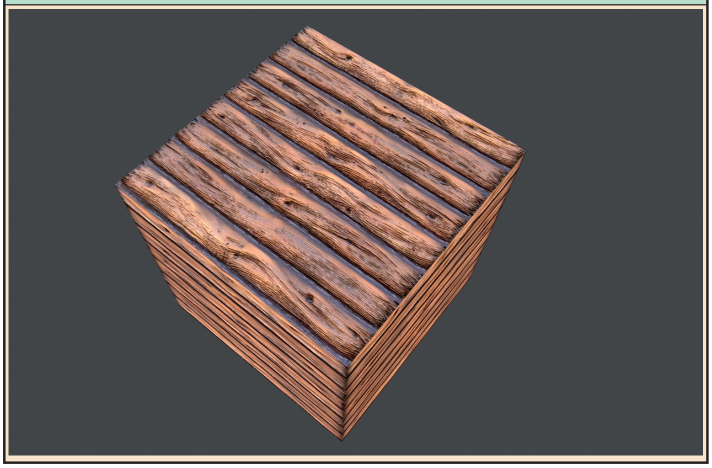


By adding a **displacement map image** to the object...

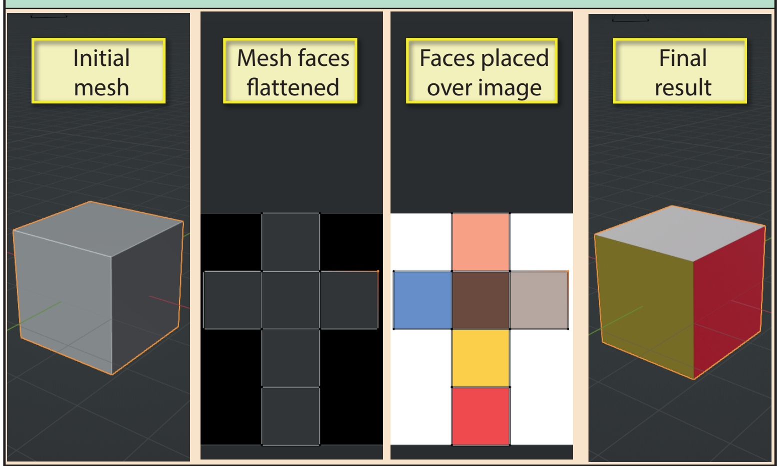


Displacement Map Image

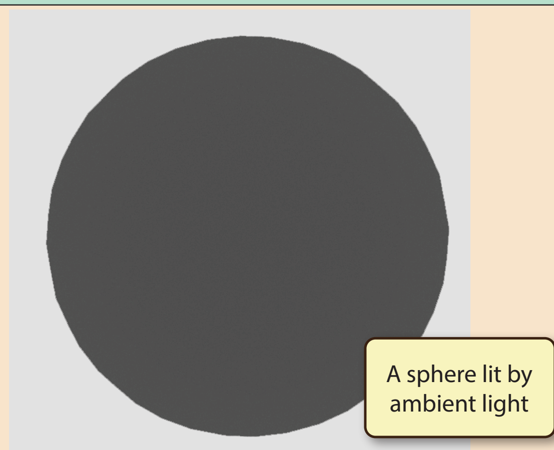
...we can increase the number of faces in the model and create genuine lumps and bumps on the surface.



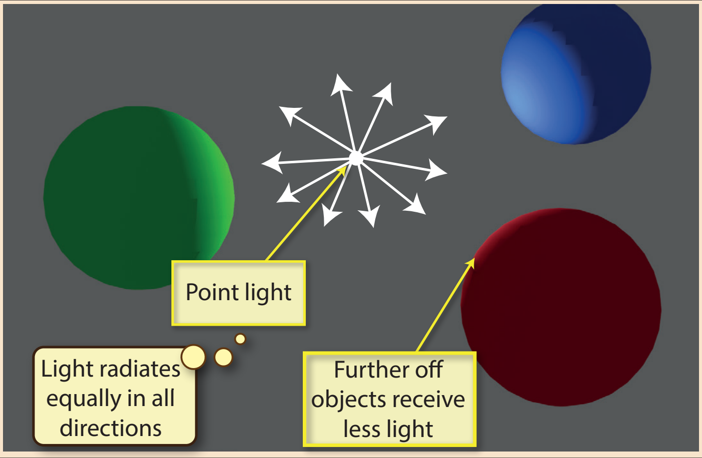
By laying out the faces of an object's mesh so that we create a flat shape, we can control which part of the texture image appears on each face (this is called **UV mapping**).



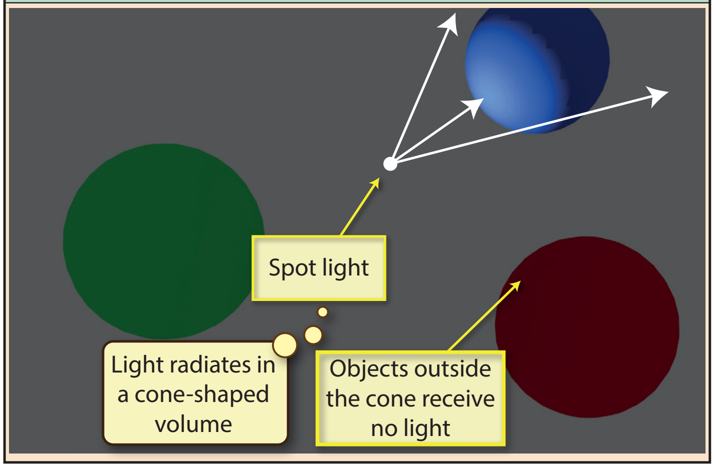
Objects in a scene are only visible because they reflect light. The most basic light is an **ambient light** in which light comes equally from all directions. This creates a flat light without shadows or highlights.



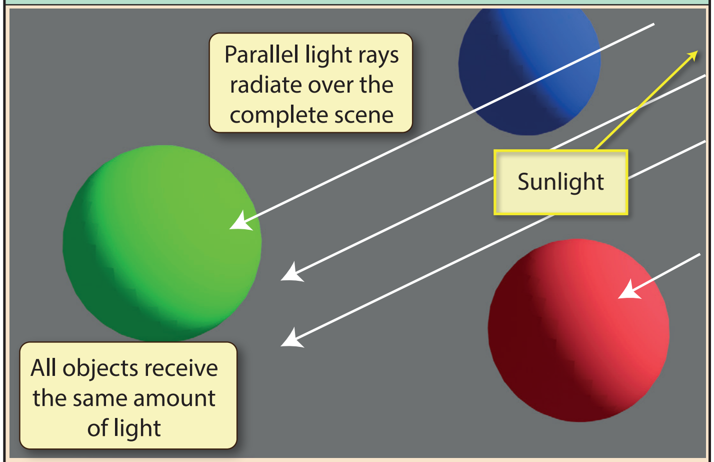
A **point light** emits light in all directions from a single point in space with the light cast on an object becoming weaker the further the object is from the light.



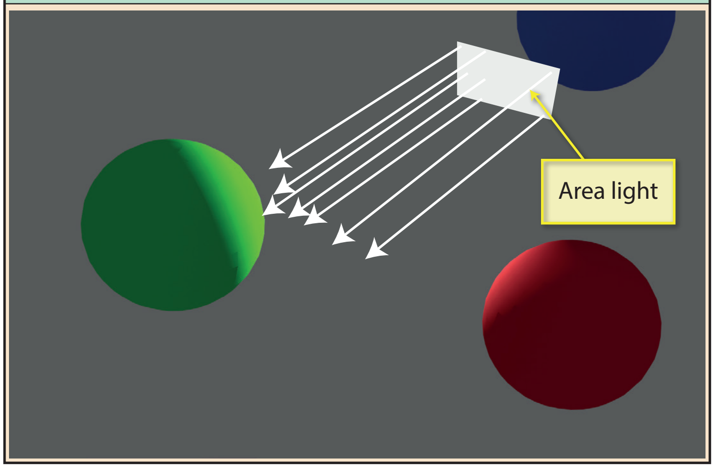
A **spot light** casts light within a cone-shaped volume (think of search light scanning the skies). Like a point light, objects at a distance receive less light.



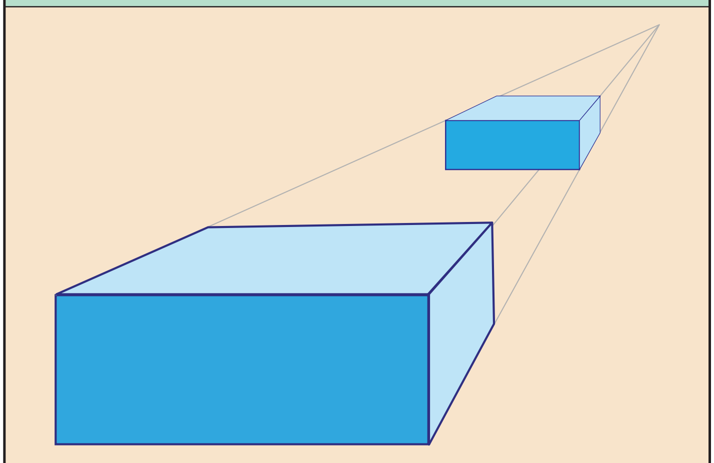
Sunlight (also called **directional light**) casts a set of parallel light rays with all objects receiving the same intensity of light. Positioning of the light is irrelevant. The angle of the light rays can be set.



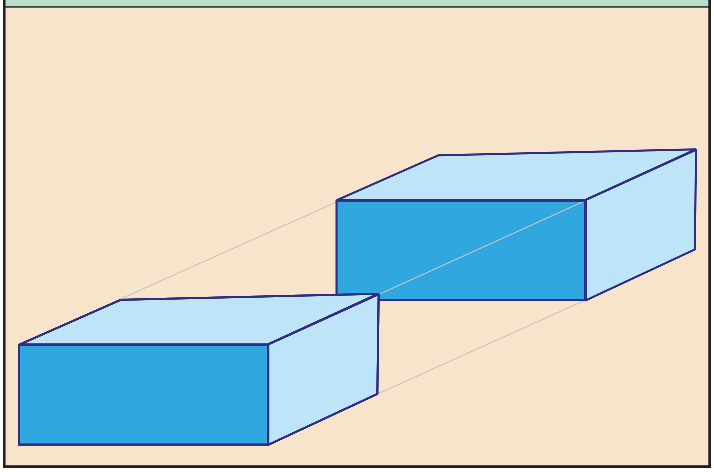
Area light simulates a light originating from a flat area such as a panel light found on a ceiling. This is a form of directional lighting.



When creating a model, we have the option to work in **perspective viewing mode** (where objects appear smaller the further away they are and real-world parallel lines converge to a single point.

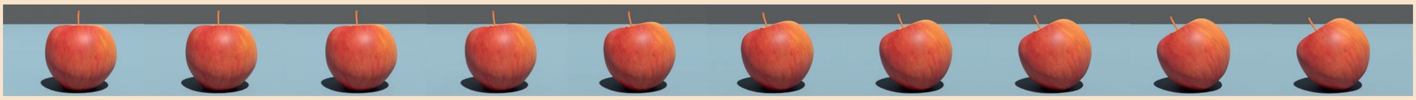


However, it is often more accurate to work in **orthographic viewing mode** where real-world parallel lines remain parallel and objects don't get smaller in the distance.

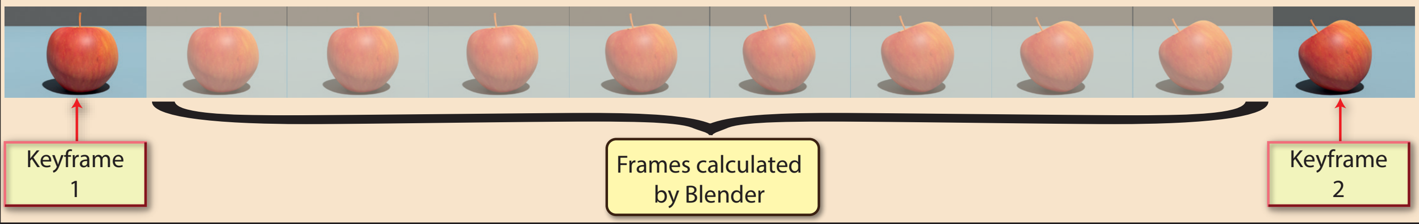


The final part of the process when creating a photographic-quality image or video animation is to **render** the model. ***Rendering*** - which is performed by a **render engine** - creates a detailed image which includes textures, shadows and reflections. The highest quality renders may take minutes or even hours depending on the processing power of the computer being used.

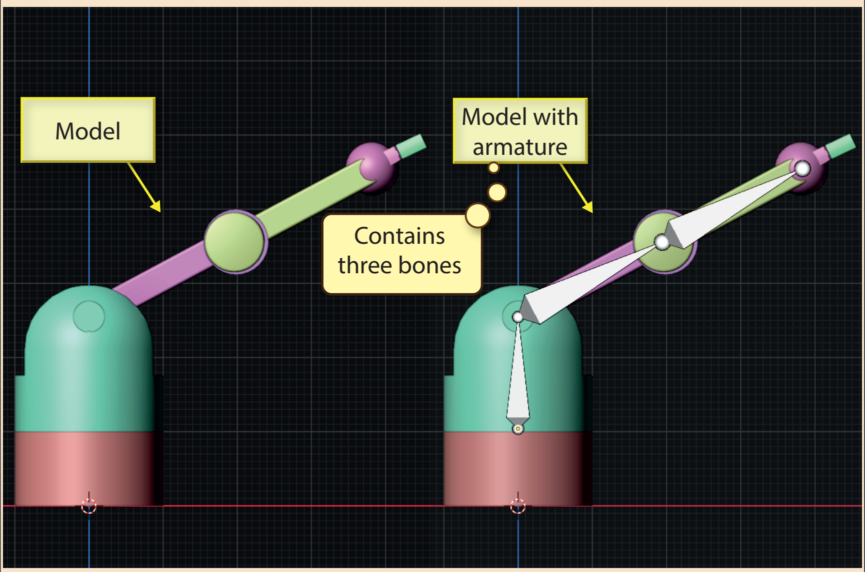
If, rather than creating a single, static image, we wish to create an animation, then we must create a set of frames, each with a slightly different image. These frames, when played in rapid succession, these create our animation.



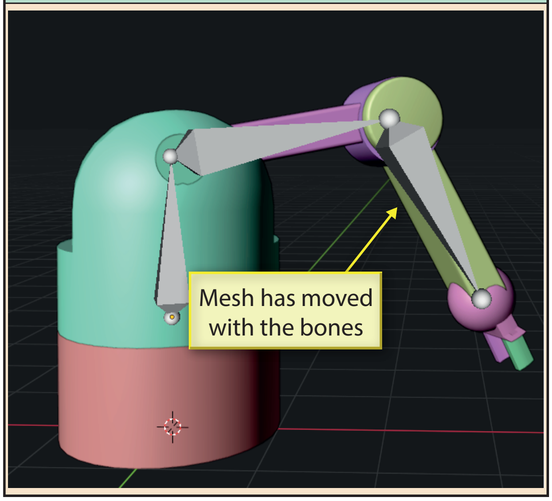
Luckily, we don't have to create each of these frames manually. Instead we create only the important frames (known as **keyframes**) and Blender automatically creates the intermediate frames.



When our animation involves reshaping meshes, then we normally add an **armature** (also known as a **rig** or **skeleton**). An ***armature*** is a set of bones and joints. Each bone is associated with some part of the mesh.



When the bones are moved, the linked part of the mesh also moves. We can now add keyframes by adjusting the bone positions and Blender will create the intermediate frames with the final animation saved as either a set of still images or as a video file.

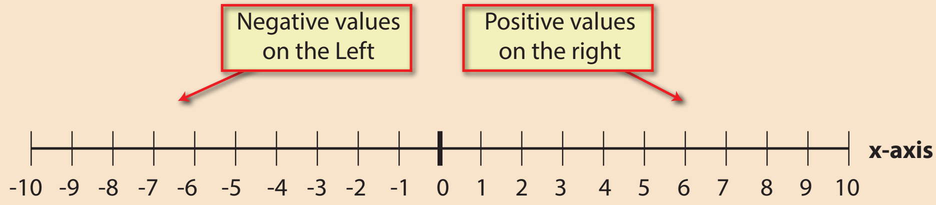


Axes and Coordinates

When creating a 3D scene, we need to specify the position of any new object.

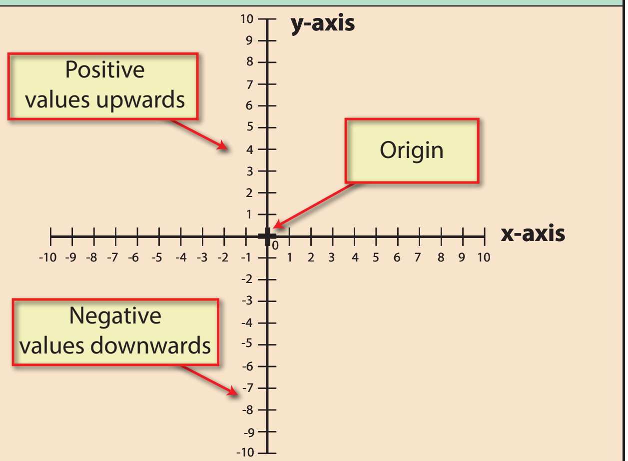
To do this we employ axes which act as a type of ruler allowing us to measure distances in one, two or three dimensions.

If we were measuring distances in a one dimensional space (say, along a telephone wire or a spider thread), we could use a single, imaginary axis (known as the **x-axis**). The centre of the axis is labelled zero with positive increments to the right and negative increments to the left.

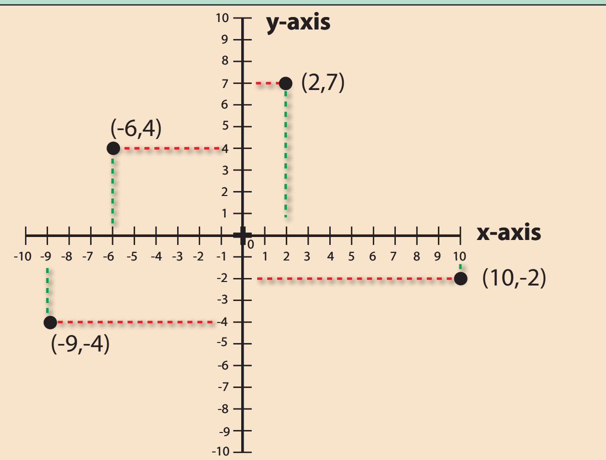


In theory, the axis expands indefinitely in both directions

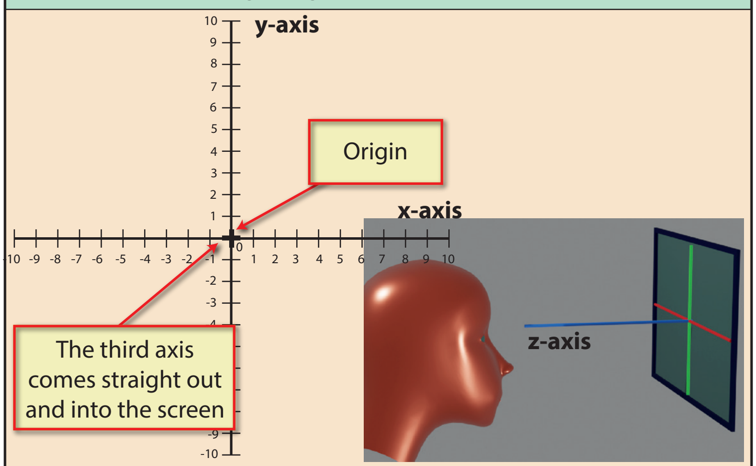
In two dimensional space – for example, a sheet of paper – we use two axes. These are perpendicular to each other and cross at their zero points. This meeting point is known as the **origin**. The new axis is known as the **y-axis**.



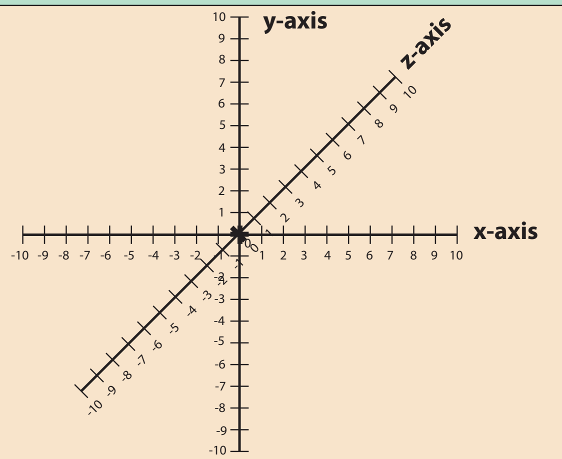
With the two axes, we can specify the position of any point in space by giving its distance along the ***x-axis*** and then along the ***y-axis***. By convention these are enclosed in parentheses and comma-separated.



For three-dimensional space (usually written as 3D space), we need three axes. The third axis (***z-axis***) is perpendicular to the other two. Of course, on paper or a 2D screen we have to settle for a mental picture of the final axis coming straight out of, and into, the paper/screen.



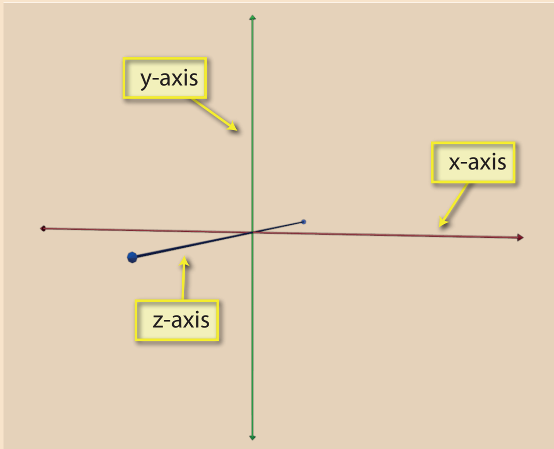
Often, in order give a visible presence to the ***z-axis***, it is drawn at 45° to the other two axes. This can lead to some confusion, initially.



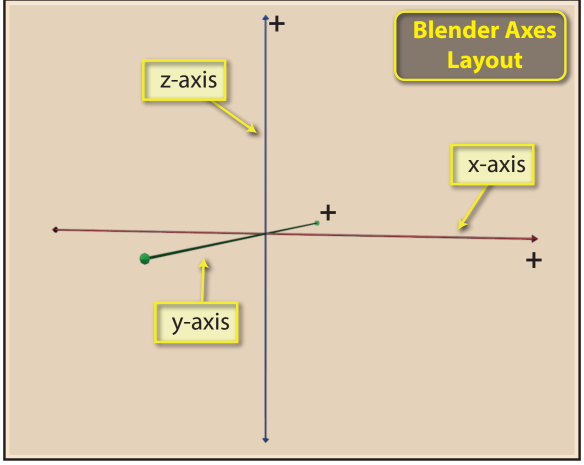
NOTE

In the last frame we saw that the z-axis had its positve direction flowing "inwards" with the negative direction flowing "outwards". In fact, there is no universally agreed convention on this and we may find that different software uses the opposite direction of flow for the z-axis.

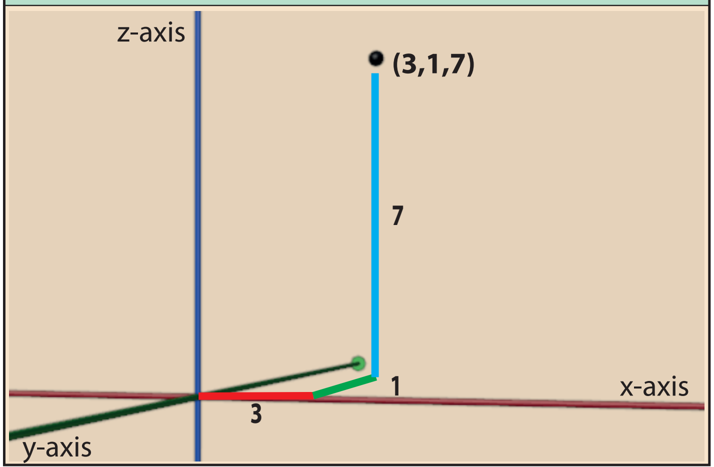
If we move our viewpoint in 3D space slightly, we can see all three axes.



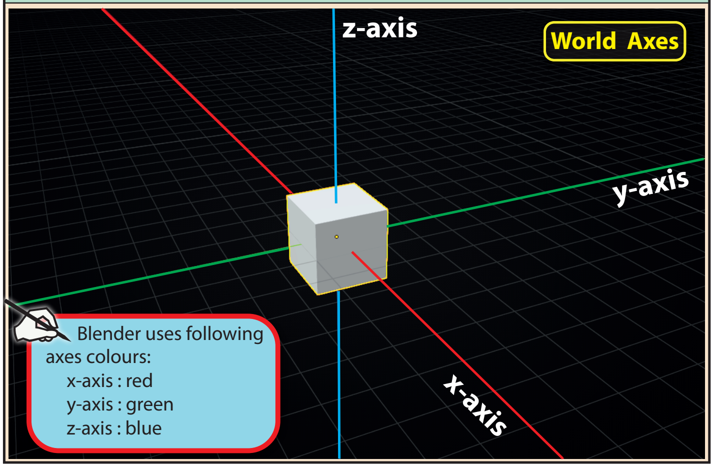
Although the previous frame gives the traditional layout of the axes, Blender rotates things by 90º, making the *z*-axis vertical. The positive end of the *y*-axis is now farthest away from our viewpoint.



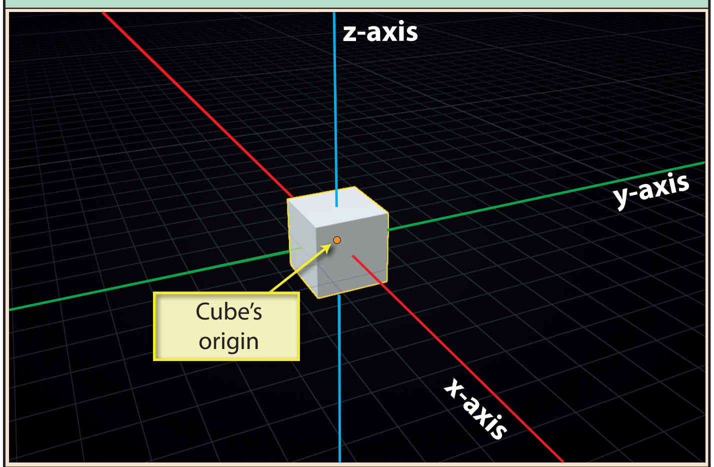
To specify the coordinates of a point in three-dimensional space we must give its distance along each of the three axes in the order *x*, *y* then *z*.



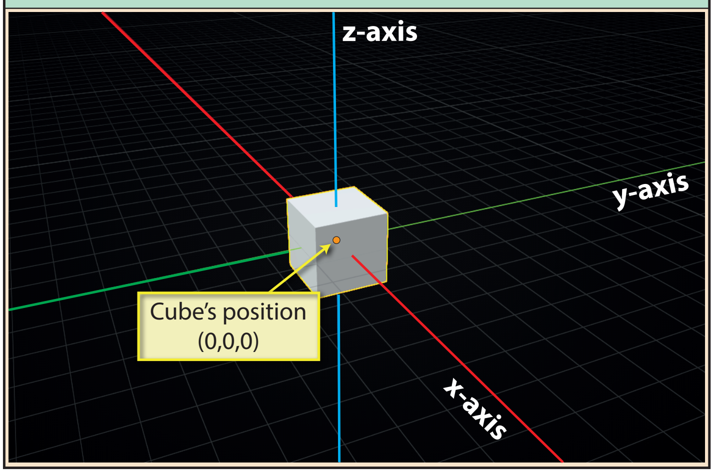
Blender uses several sets of 3D axes. The most important of these is the **World Axes**. It is this set of axes that is used to specify the position of every object we add to a Blender scene. Only the *x* and *y* axes are visible initially (the *z*-axis has been added below).



In a Blender scene, an object's centre (known as the **object's origin**) is used when specifying the position of the complete object. This position is represented in Blender by a **small orange circle**.



Since the ***Cube's origin*** is at the ***World Axes's origin***, the Cube's coordinates are given as ***(0,0,0)***.



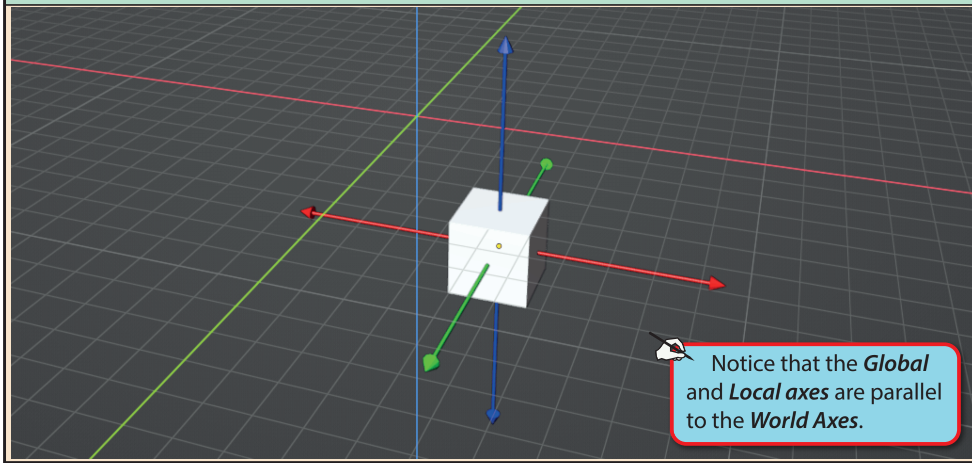
Every object in Blender maintains several different sets of axes, although, unlike the ***World Axes***, the other axes cannot be made visible.

Two important sets of axes are the **Global Axes** and **Local Axes.** Every object in a Blender project has its own set of ***Global*** and ***Local*** axes.

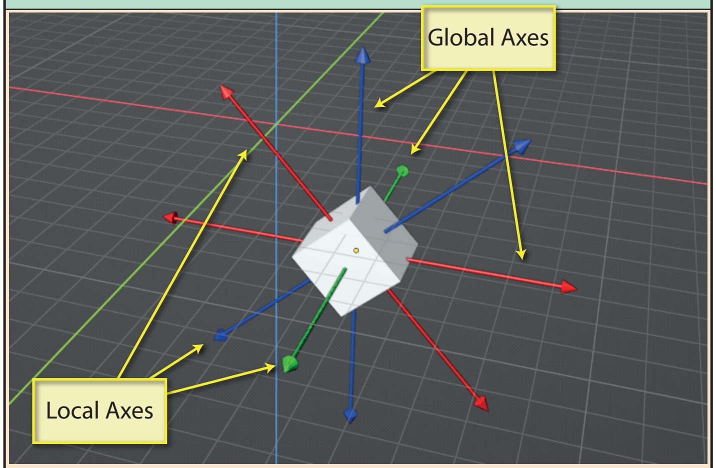
A less important set of axes is the **View Axes** whose position is determined by the location from which we are viewing the scene.

These three axes types are explained briefly here, but we will encounter other axes later in the text.

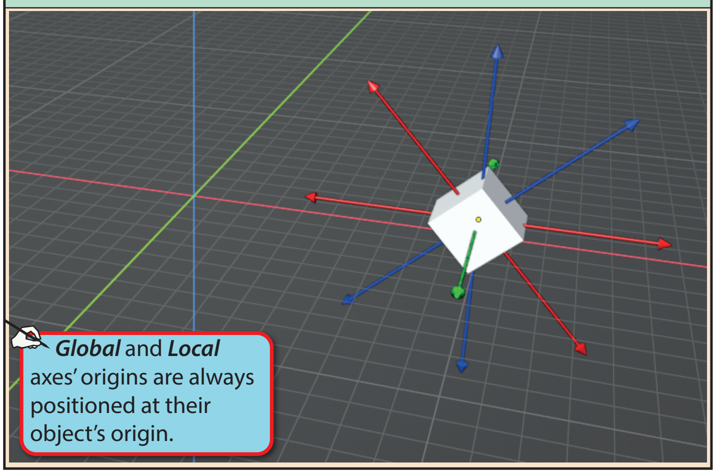
In the scene below we have a ***Cube*** object which has been moved away from its original position at the ***World Axes origin***. Drawn onto the ***Cube*** are its normally invisible **Global** and **Local axes**. These two sets of axes initially occupy the same position and so only one set of axes are shown below centred on the ***Cube's*** origin.



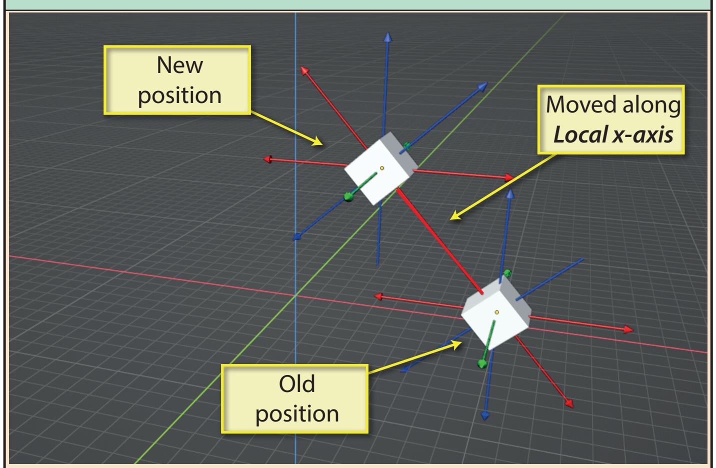
If we rotate the ***Cube***, the ***Global Axes*** remain parallel to the ***World Axes*** while the ***Local Axes*** rotate with the ***Cube***. Notice that the ***Global*** and ***Local*** *y*-axis (in green) still lie along the same path since it is that common axis that the ***Cube*** has been rotated



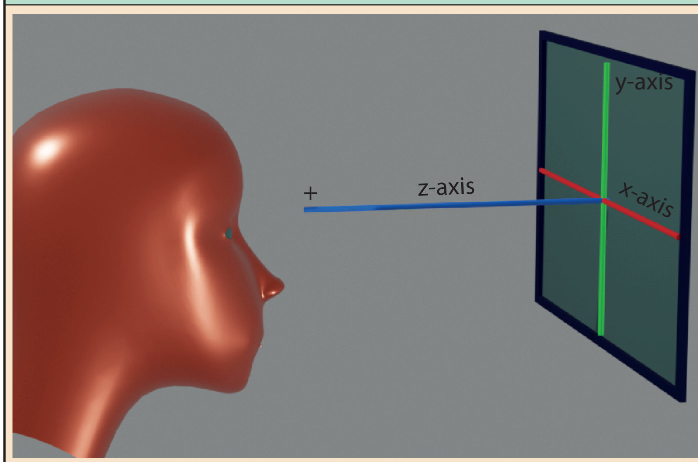
If we move the ***Cube***, both the ***Global*** and ***Local axes*** will move with it, maintaining their origins at the centre of the ***Cube***.



As we'll see later in the text, we can use any of these axes - **Global** or **Local** (or any other axes set) **-** when moving, rotating or resizing objects. Below, we can see the result of moving our previously rotated ***Cube*** along its ***Local x-axis***.



An object's **View axes** have their orientation based on the screen's surface. The *x* and *y* axes are orientated in the traditional maths layout, *x* being horizontal and *y* vertical with the positive end of the *z*-axis coming out of the screen.

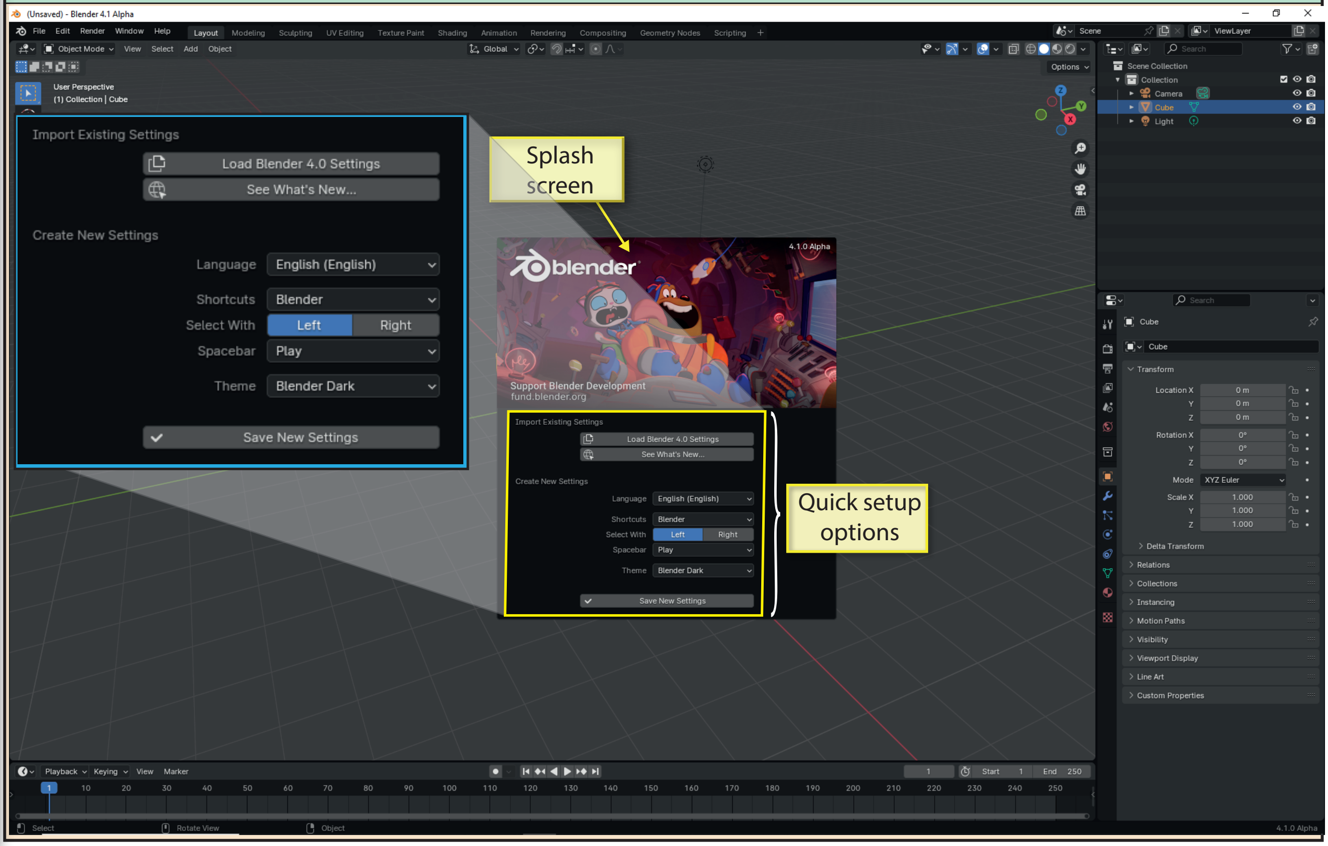


STARTING BLENDER

In this section we'll identify the main elements of the Blender interface and have a closer look at how to use the ***3D Viewport*** where we will do most of the work needed to create the basic layout of the scene we are working on.

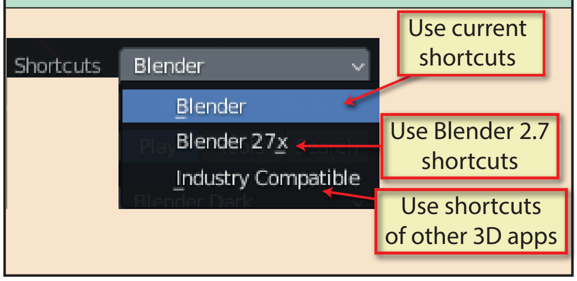
Starting Blender

The very first time we open Blender it displays a splash screen image in the centre of its window. Look carefully, because this is the one and only time it will look this way. In the bottom half of the splash screen there's a **Quick Setup** option.

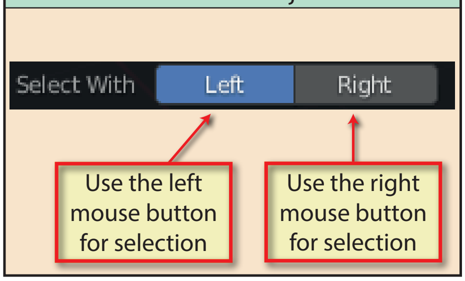


Load Blender 4.0 Settings, allows us to install the settings we may have created in our previous version of Blender. **See What's New...** is a link to the web page showing the new features in this version of Blender. **Language**, allows us to choose which language is displayed in all Blender menus and messages.

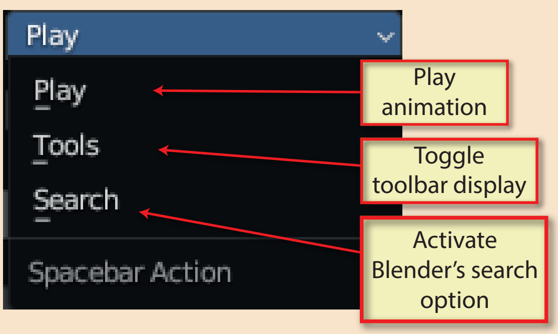
Shortcuts allows us to select which keyboard shortcut standard we wish to use.



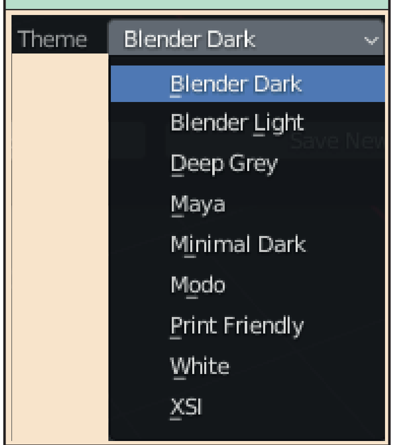
Select With determines which mouse button is used to select objects in a scene.



Spacebar sets which action is to be performed when the spacebar on the keyboard is pressed.

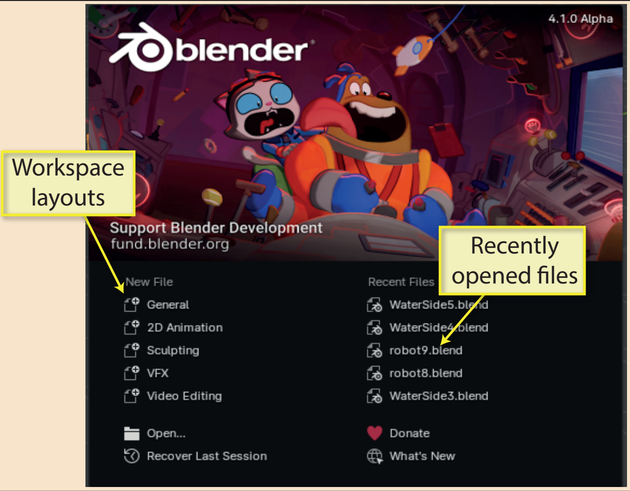


Theme selects the "look" (or theme) of the Blender window elements.



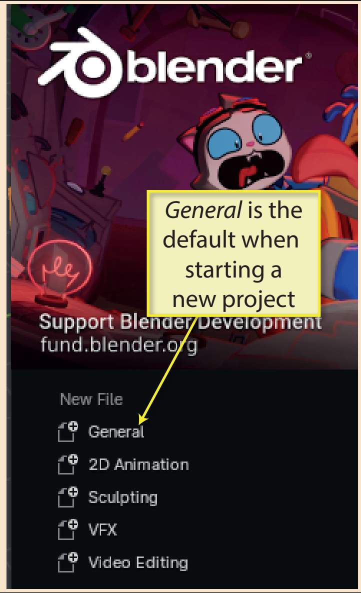
Save New Settings saves the options we have selected here, using them in this and future projects.

The next time you load Blender, the ***Splash screen*** will have changed. Now it shows mainly *file open* options including the names of recently opened files and the *screen layout* options.

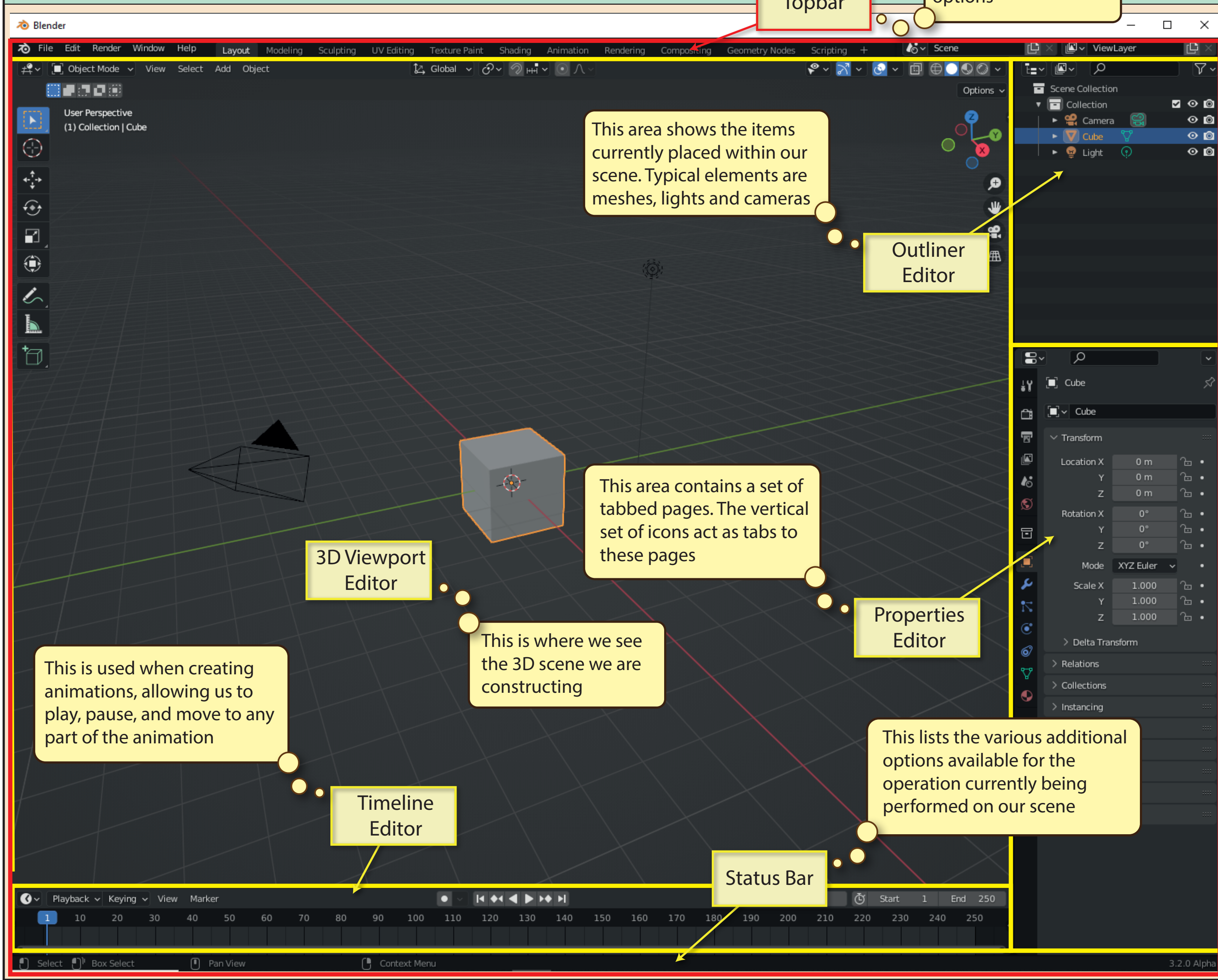


When we start a new Blender project, chances are we'll want to use the **General** layout which is the default option.

We can do this by left-clicking on **General** in splash screen, selecting ***File>New File>General*** from the main menu, or simply by clicking anywhere outside the splash screen.



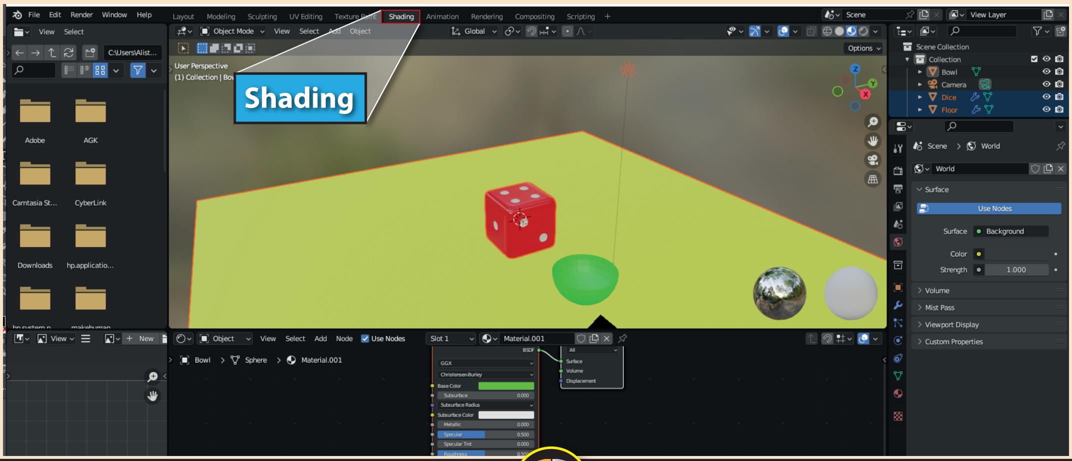
Having selected *General*, the splash screen is removed and we are presented with the layout shown below. The layout is divided into **regions**. The various regions are labelled below.



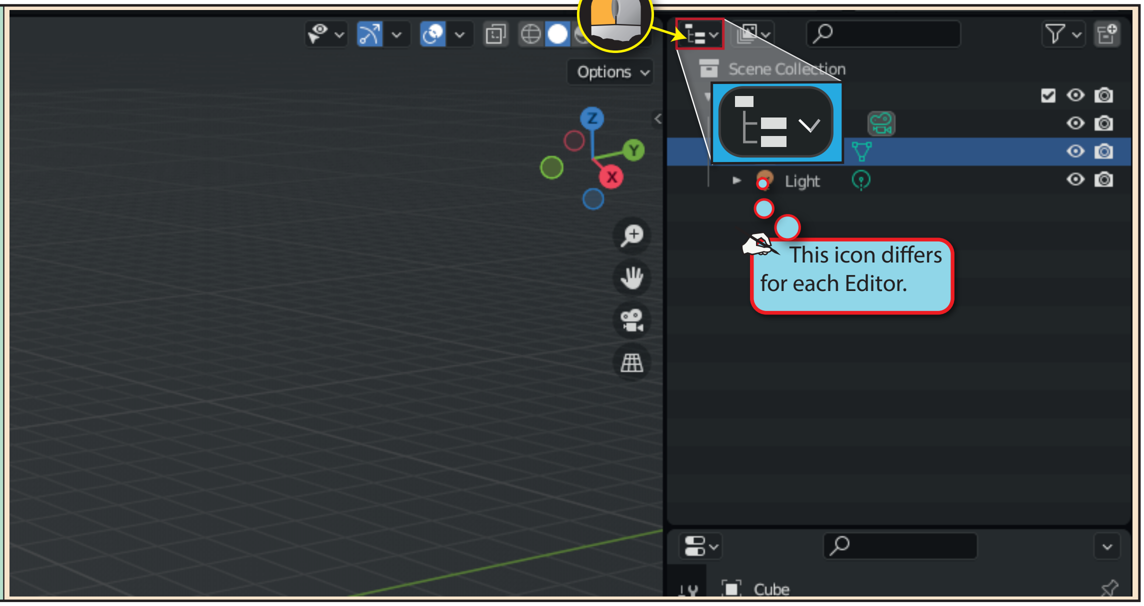
Workspace Adjustments

Before discussing the various parts of the Blender interface in more detail, we'll take a moment to see how the layout of the Blender window can be modified.

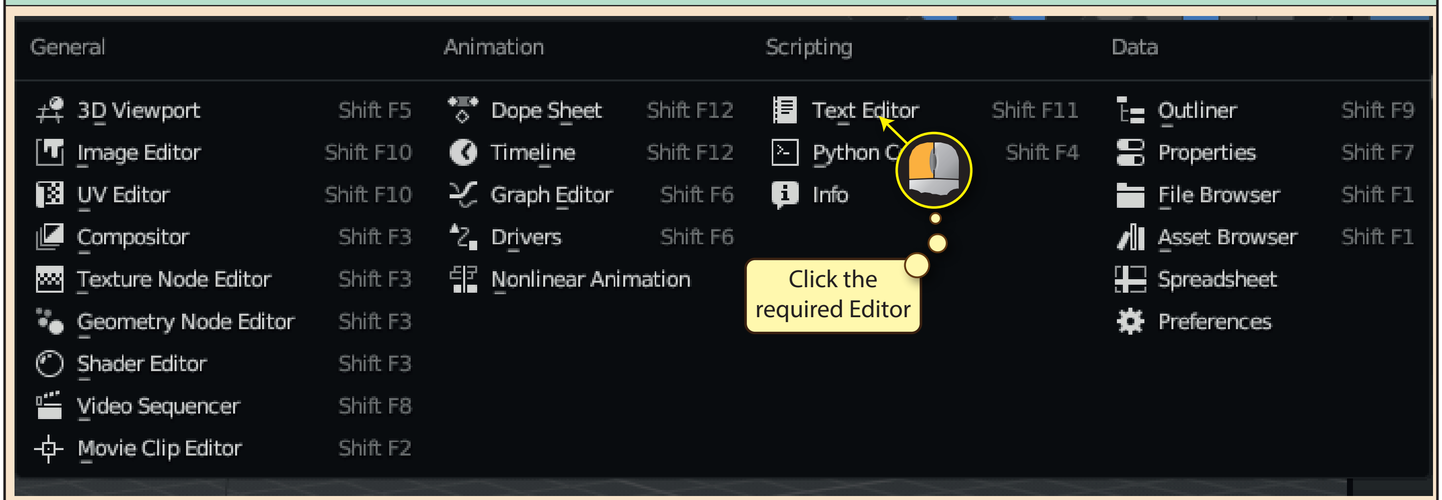
One of easiest ways of adjusting the Blender workspace is by selecting a new layout from the options available in the ***Topbar***. Below we can see part of the layout created by the ***Shading*** option.



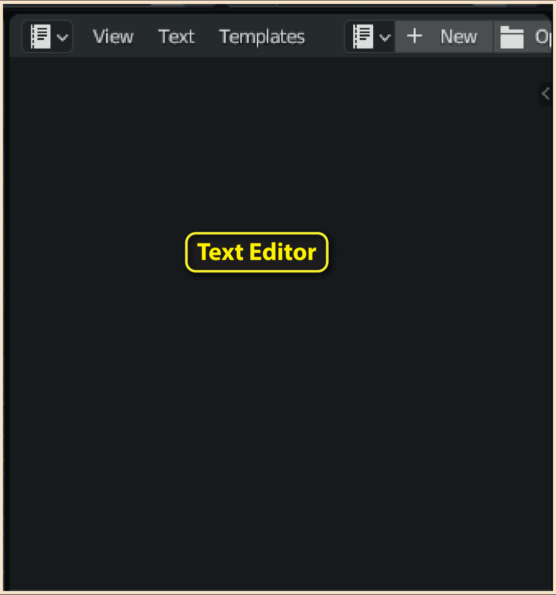
But we can also change individual elements of the layout. For example, any Editor within the window can be changed to a different Editor by first clicking on the symbol in the top left of the Editor to be changed. Here, the symbol in the ***Outline Editor*** has been selected.



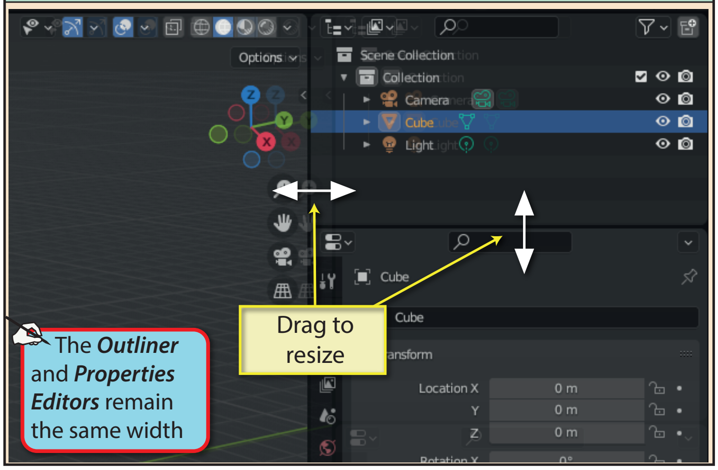
Once we click on the icon, a large table of Editor options appear. From the options given here, we can change the Editor displayed in that area of the Blender window.



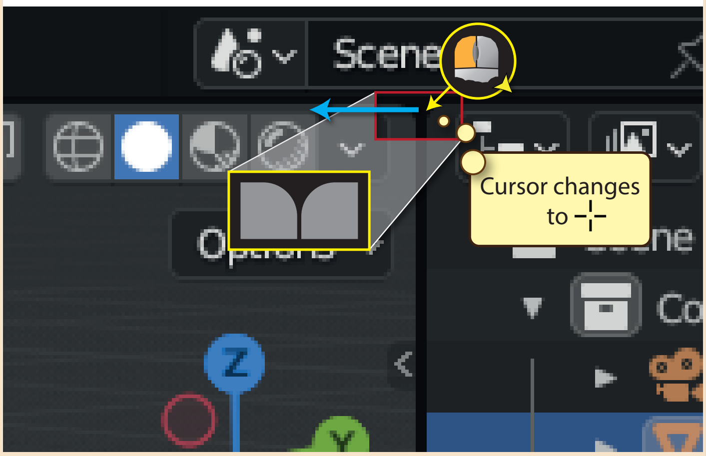
The new Editor will occupy exactly the same space as the previous one.



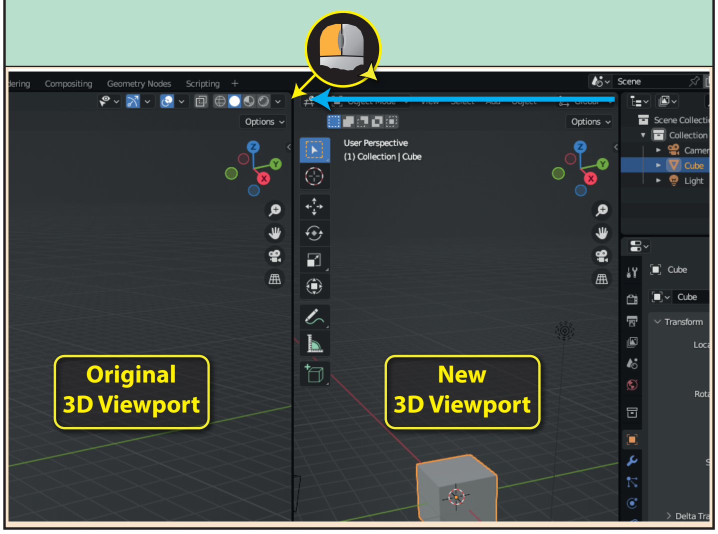
Each Editor window can be resized by dragging on an edge.



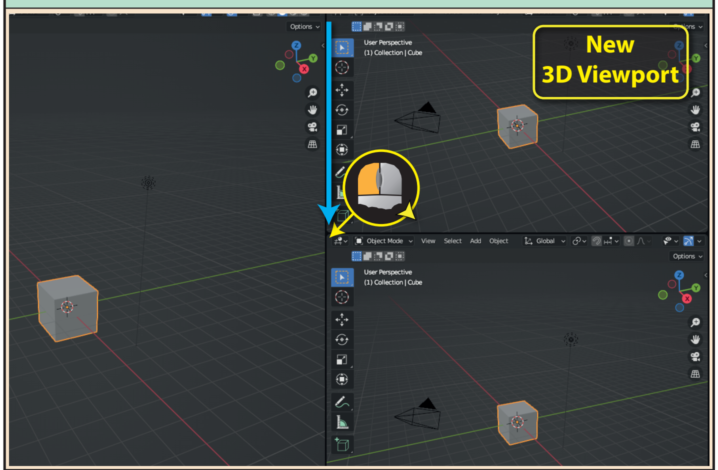
We can even split an Editor area into two separate areas by dragging on the hard-to-see curved boundary displayed in every corner of all Editors. The mouse cursor will change to crosshairs if over the correct position.



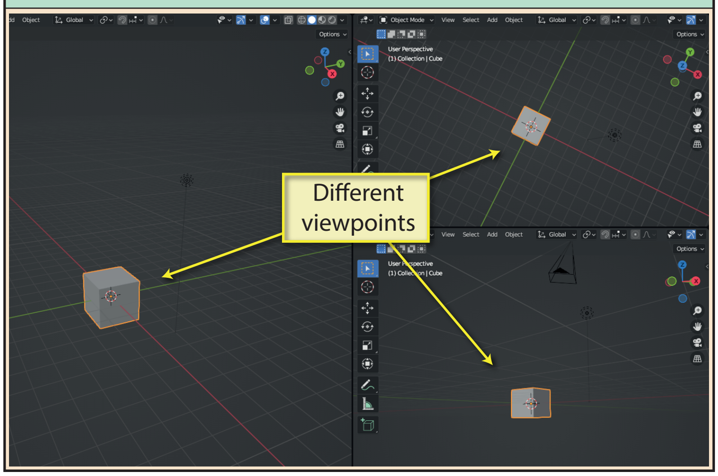
Below we can see the result of dragging the curved edge between the ***3D Viewport*** and ***Outliner Editor*** to the left.



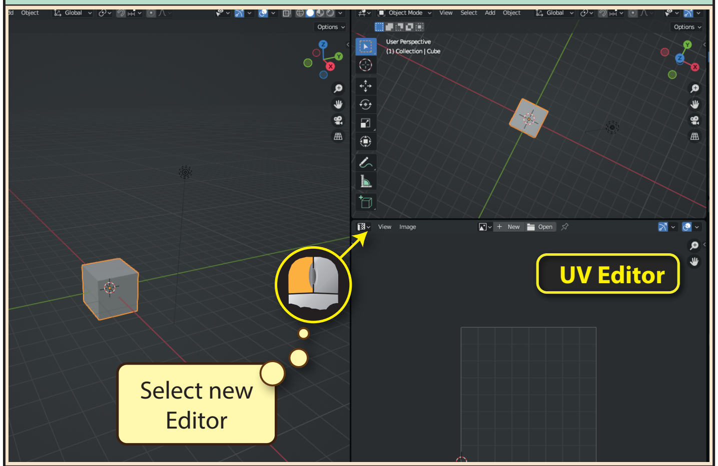
If we drag vertically, the area will split horizontally.



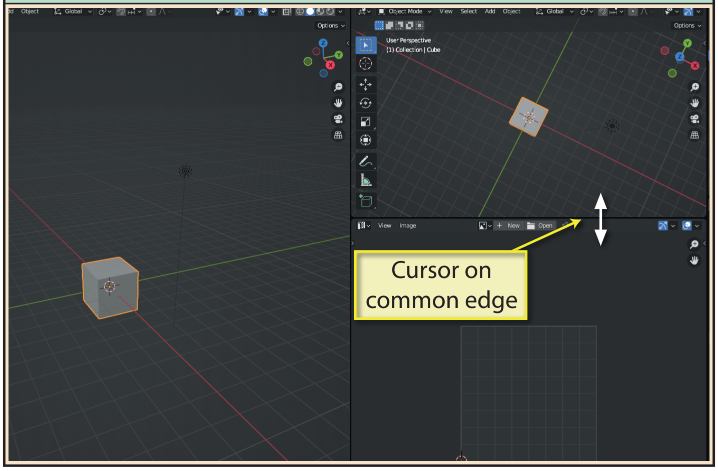
Each area is completely independent, so, we could, for example, have different viewpoints in each area (we'll see how to change viewpoints shortly).



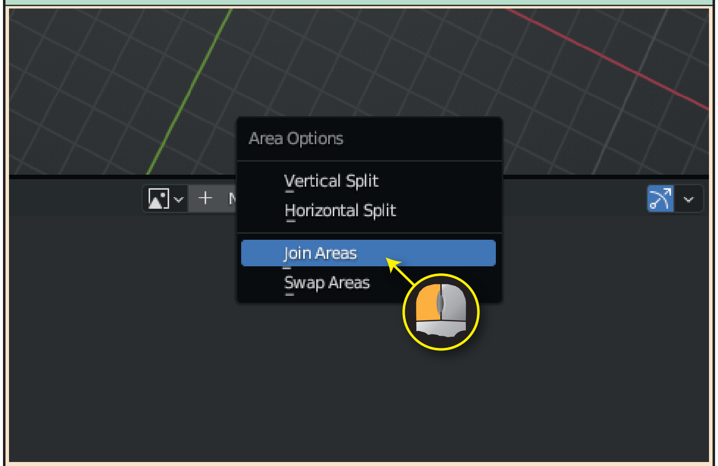
Since each newly created area is separate and independent, we are free to change the Editor appearing in that area using the same approach as we saw earlier.



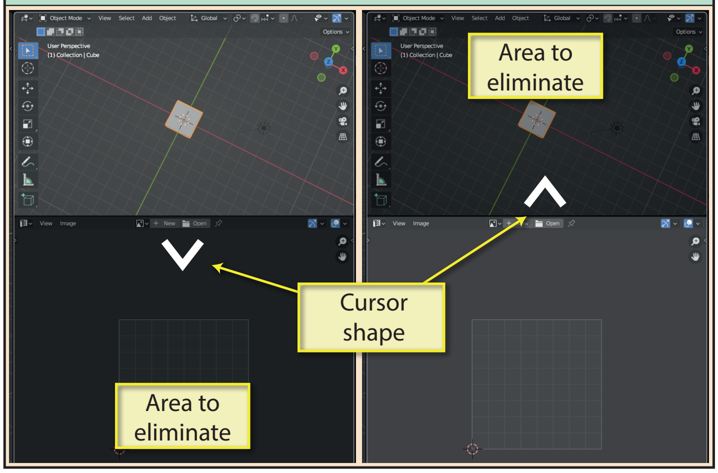
The simplest method of merging two areas into a single area is to start by moving the cursor over the common edge between the Editors - where it changes to a double arrowed line.



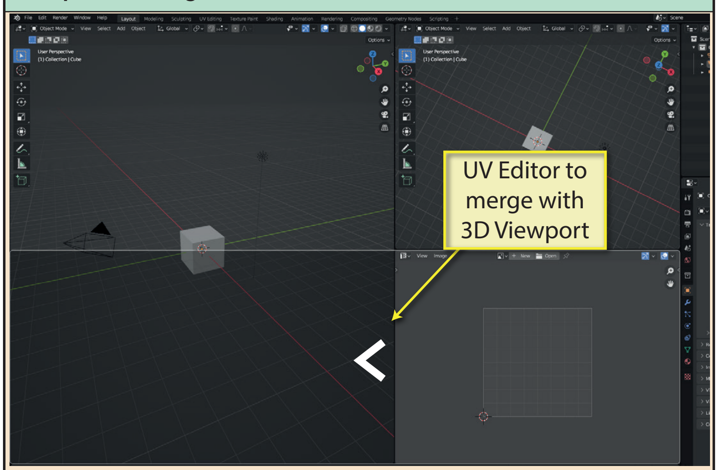
Right-clicking the mouse at this point produces a popup menu from which we need to select ***Join Areas***.



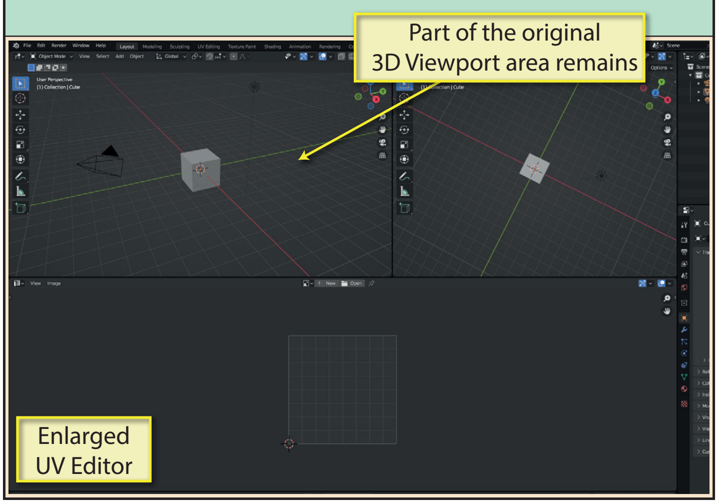
Now we need to move the mouse into the area we wish to eliminate. The two options in our example are shown below.



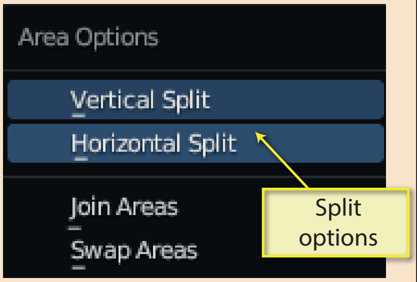
Note that, where the common edge between the two areas to be merged are of a different length, additional area splitting will occur. For example, if we try to merge the ***UV Editor*** to the ***3D Viewport*** to its right...



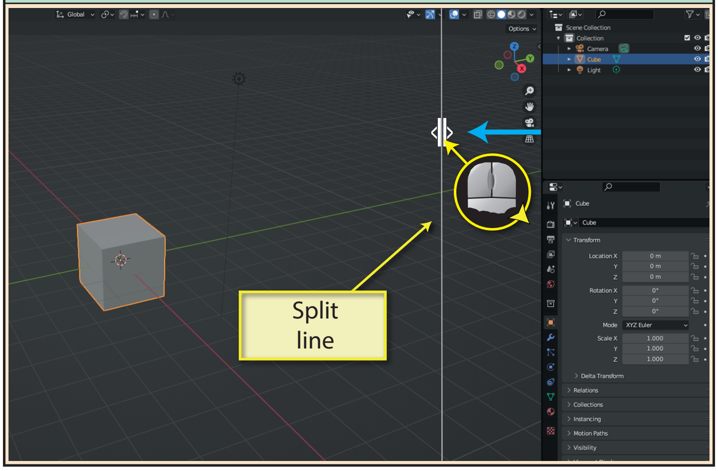
...the ***3D Viewport*** splits to ensure that the new area has the same height as the ***UV Editor***.



We can also see that the popup menu not only offers a merge option, but also offers two split entries. These can be used as an alternative way of splitting an area into two.

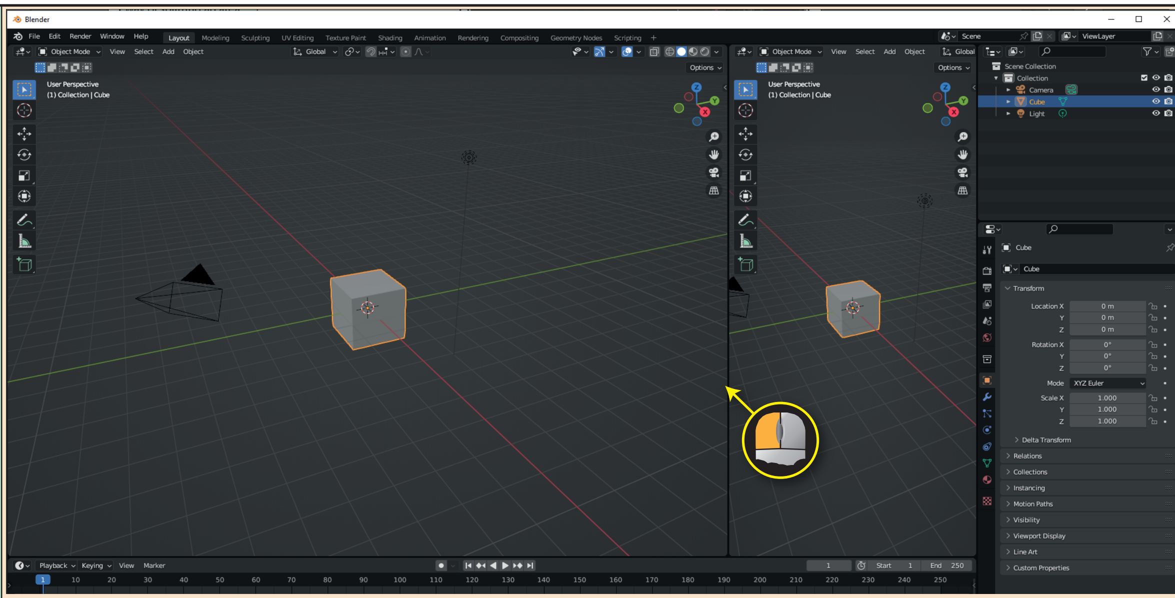


After selecting one of these options (***Vertical Split*** in the example below), a vertical or horizontal line appears (which depends on the menu option selected). We can drag this to define the width/height of the new area.

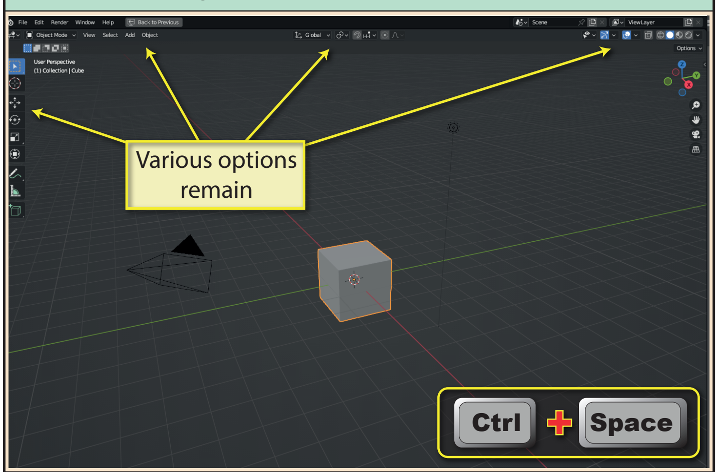


To finalise the size of the new area, we need to click the left mouse button.

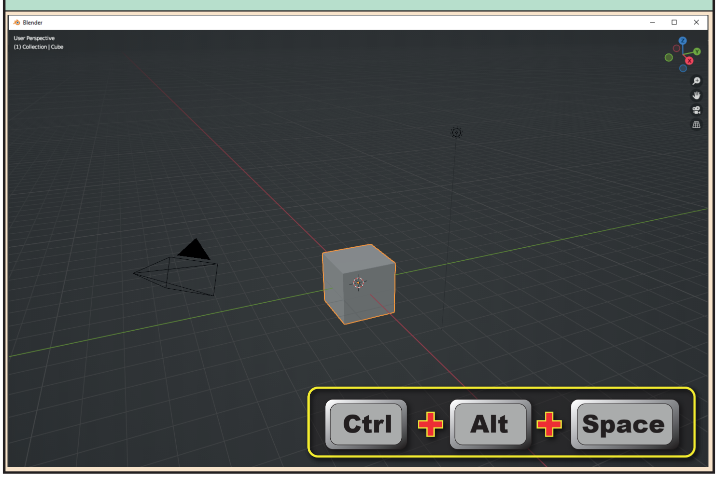
The resulting new area is shown here.



When we are working in a specific area of the Blender window, we can expand that area to occupy the full window by pressing one of two key sequences. Pressing **Ctrl+Space** when working in the ***3D Viewport***, gives the result shown below.



Pressing **Ctrl+Alt+Space** fills the Blender window completely with the scene being constructed.
Press the same key combination (**Ctrl+Space** or **Ctrl+Alt+Space**) to return to the previous layout.

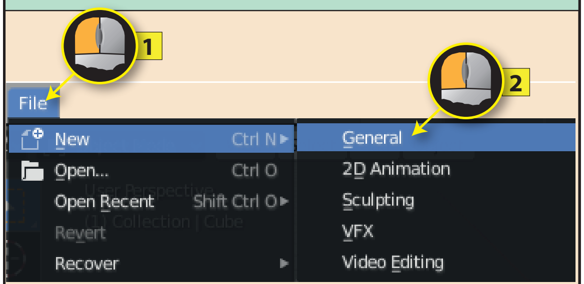


Workspace Details

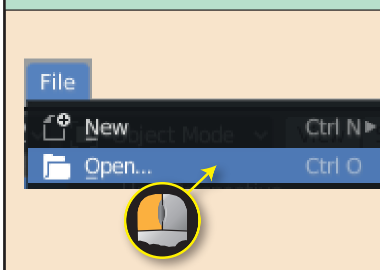
The **Topbar** contains many items including the ***main menu*** bar and a set of ***workspace layouts*** which we can switch between as we progress through the various stages required to create our model.

The **main menu** contains several categories but at this point, we will look briefly at only some of the ***File*** menu options.

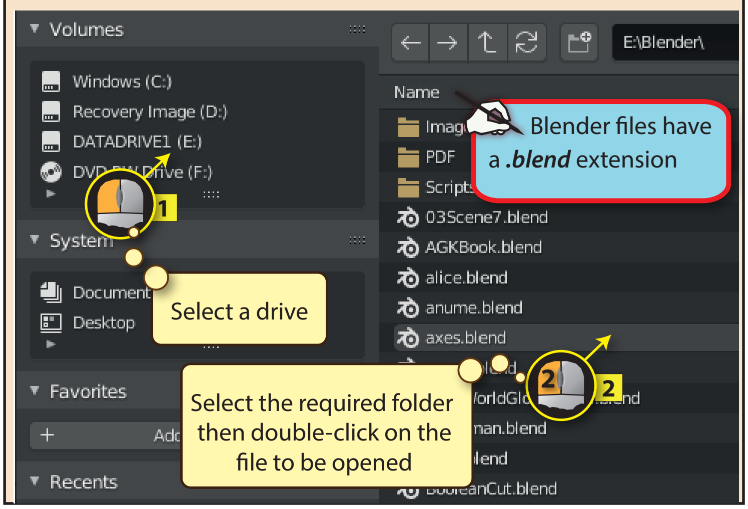
To start a new project we need to select the ***File|New*** option then select ***General*** from the submenu.



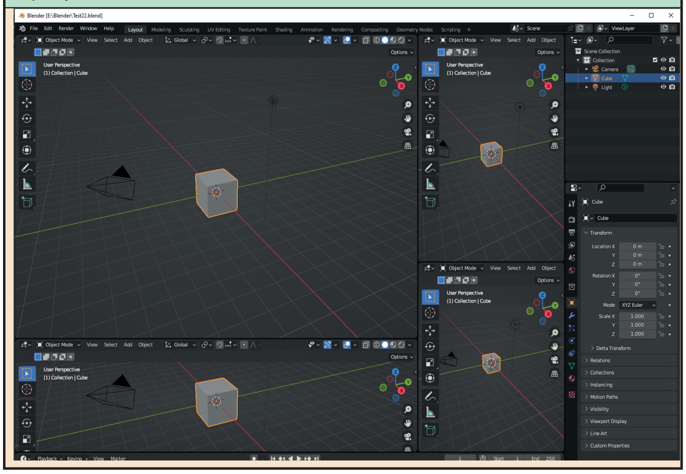
To load an existing project, we use the standard ***Open...*** option.



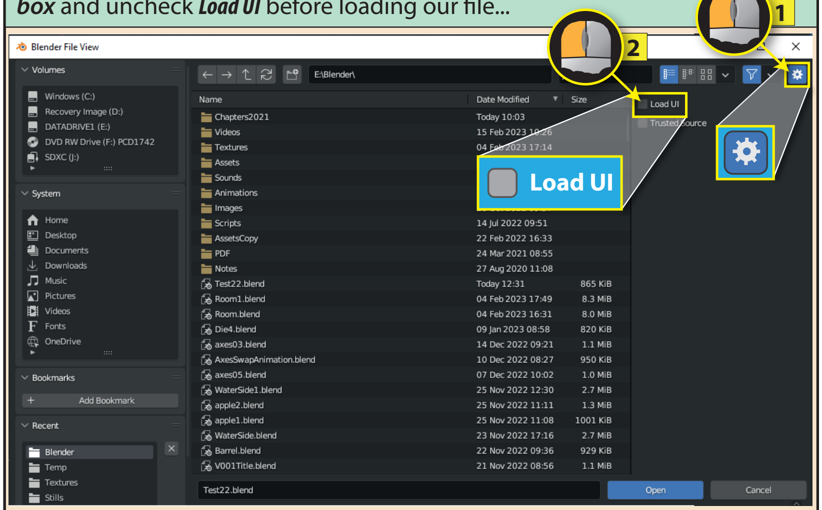
Open... brings up the window shown below. Blender has its own format for selecting drives, folders and files.



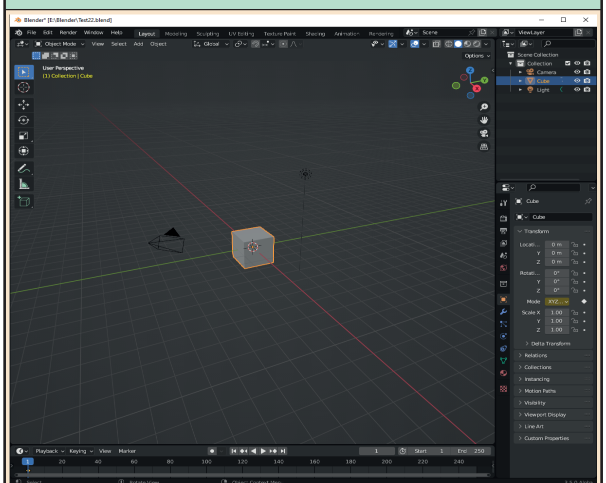
When Blender saves a project, it saves not only the scene we have been creating but also the current layout of the Blender window. For example, if we had the layout shown below when a project is saved...



...then, by default, it would later load the file with exactly the same layout. However, if we click on the **Settings icon** in the ***Load File dialog box*** and uncheck ***Load UI*** before loading our file...

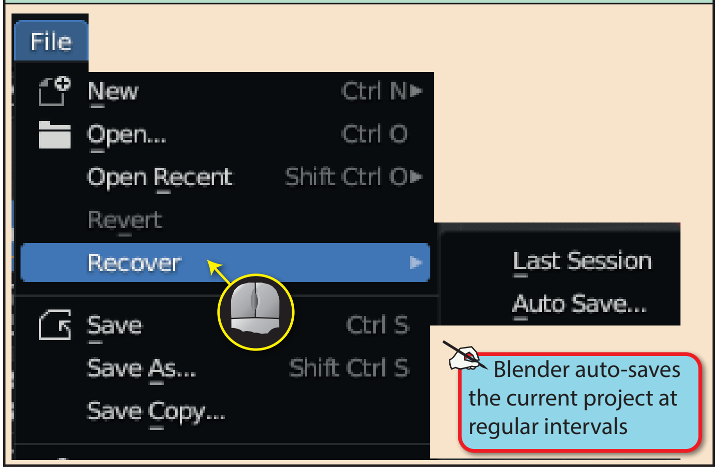


...then only the scene will be loaded and the layout of the Blender window will remain unchanged from its current state.

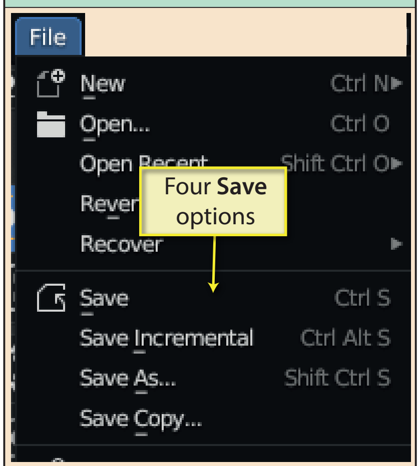


To open a recently accessed project, move over ***Open Recent...*** then select from the list of projects presented.

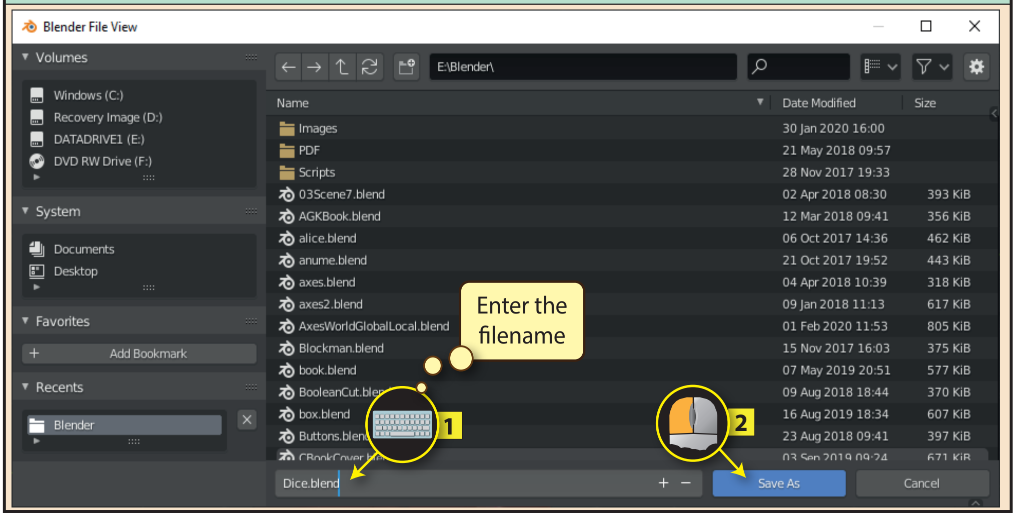
Recover offers two options. **Last session** reloads a file named ***Quit.blend*** which is saved automatically when Blender is exited. **Auto Save** allows a previously auto-saved file to be reloaded.



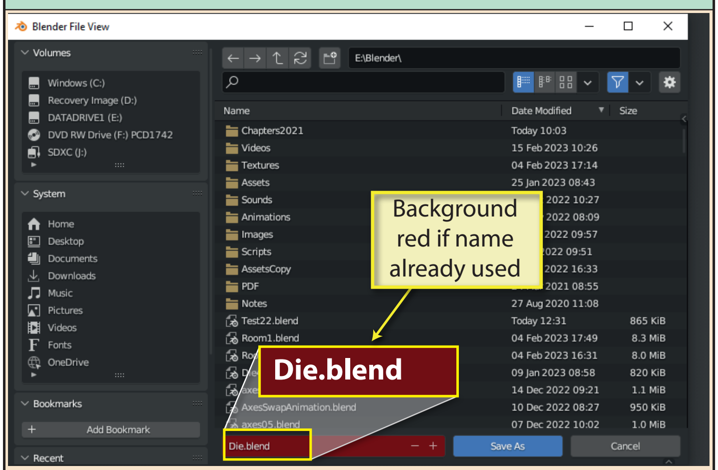
To save the current project select ***Save*** **,** ***Save Incremental*****,** ***Save As...*** or ***Save Copy...*** from the ***File*** menu.



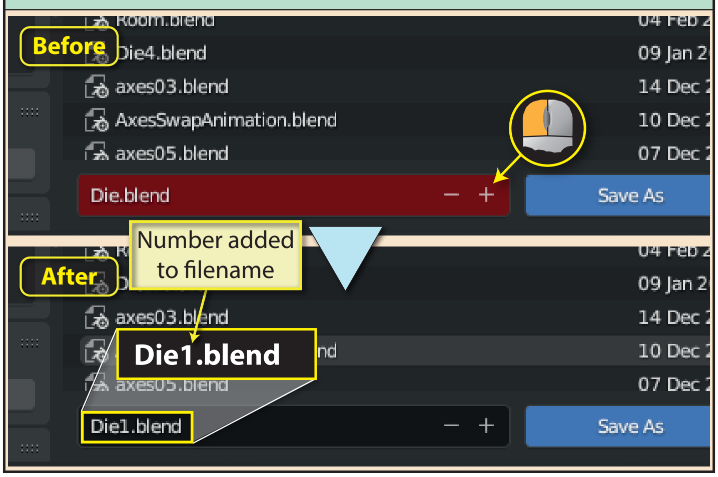
On the first save, we need to use ***Save As...*** , then select the drive and folder before entering the filename for our model and pressing the **Save As** button.



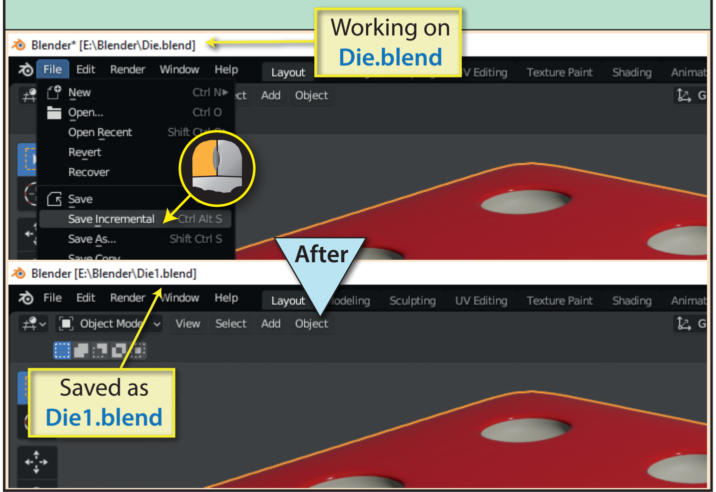
If the filename matches an existing filename, the name will appear with a red background.



To save a file as a later version of an existing file, press the **+** symbol to the right of the filename. This adds an incrementing numeric to the end of the name.



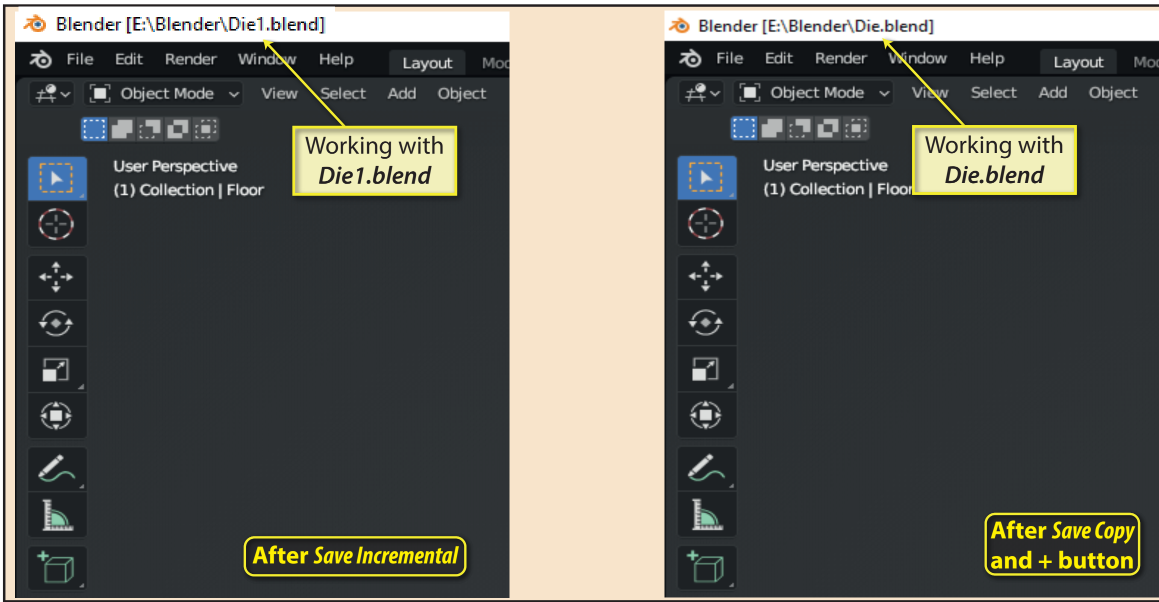
Save Incremental performs the same action as pressing the **+** button does when we've used ***Save As***. In the example below, **Die.blend** is saved using ***Save Incremental***. This creates the new file **Die1.blend**.



Save Copy at first appears to produce exactly the same results as ***Save As***, but there is a subtle difference between it and the second option.

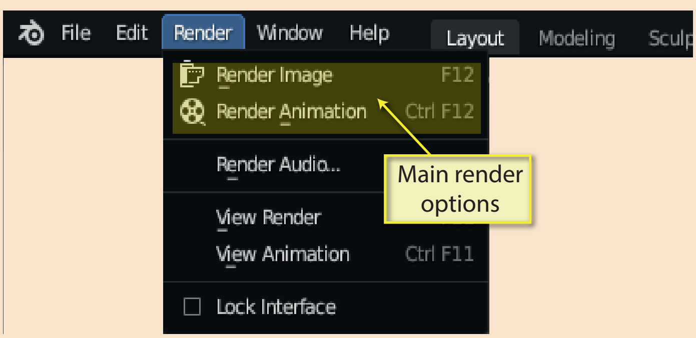
Let's assume we are working on a file called ***Die.blend*** and then choose the ***Save Incremental*** option (or ***Save As*** and the **+** button) which names the new file ***Die1.blend***. If we continue to work on the scene currently showing in the **3D Viewport**, we will be working with the file ***Die1.blend***.

However, if we use ***Save Copy*** on our ***Die.blend*** project and press the **+** button to save the file as ***Die1.blend***, before continuing to work on the scene, we would be working with the original file, ***Die.blend*** and ***Die1.blend*** would be saved as a backup storing a copy before any new changes were added.

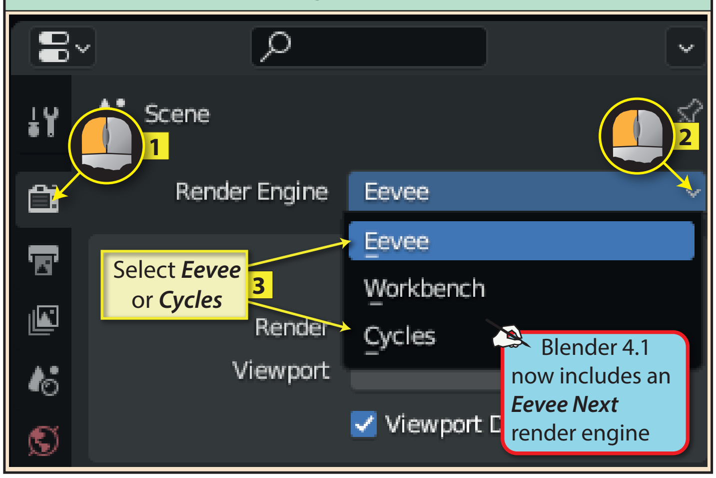


Save will use the same filename as the current project and so overwrite the previously saved copy.

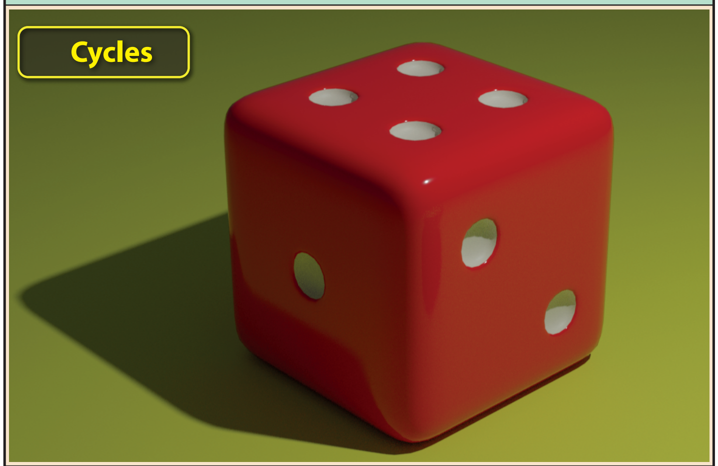
Another main menu option that will prove useful later is ***Render***. The two main options here are ***Render Image*** (used to create a still image) and ***Render Animation*** (used to create an animation).



When we select ***Render Image*** the rendered scene appears in a separate window. When rendering, we can choose between the **Eevee** and **Cycles** render engines. This choice is made in the ***Properties Editor's Render page***.

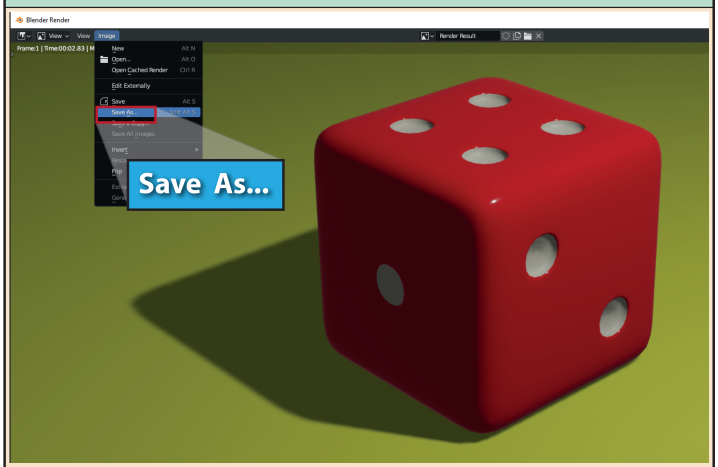


The **Cycles** render engine gives the more accurate result, but may take some time to arrive at the final image.

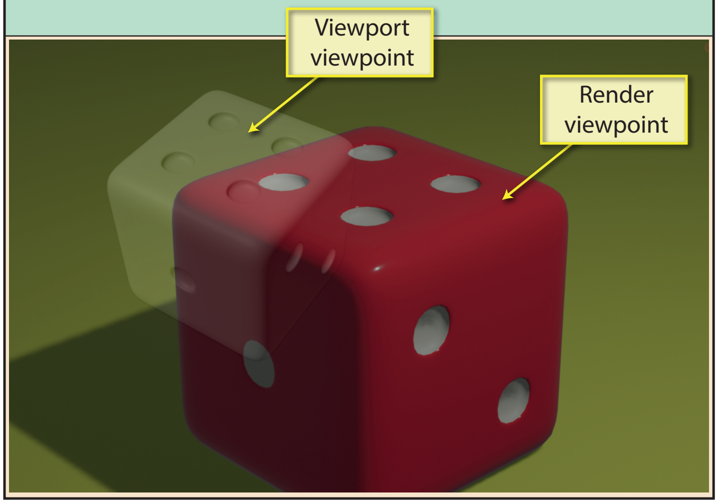


The **Eevee** engine is much quicker, but gives less accurate results.

When we select ***Render Image*** the rendered scene appears in a separate window and this new window's menu allows us to specify where the image is to be saved.

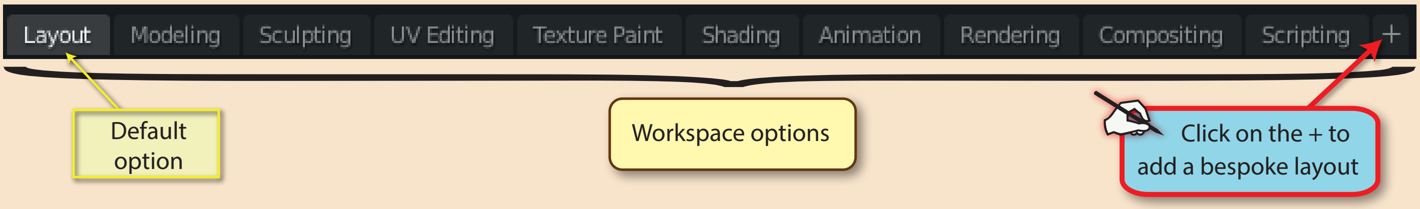


The render image's viewpoint is determined by the **Camera** object in our scene and this will almost certainly be different from the viewpoint used in the **3D Viewport**.

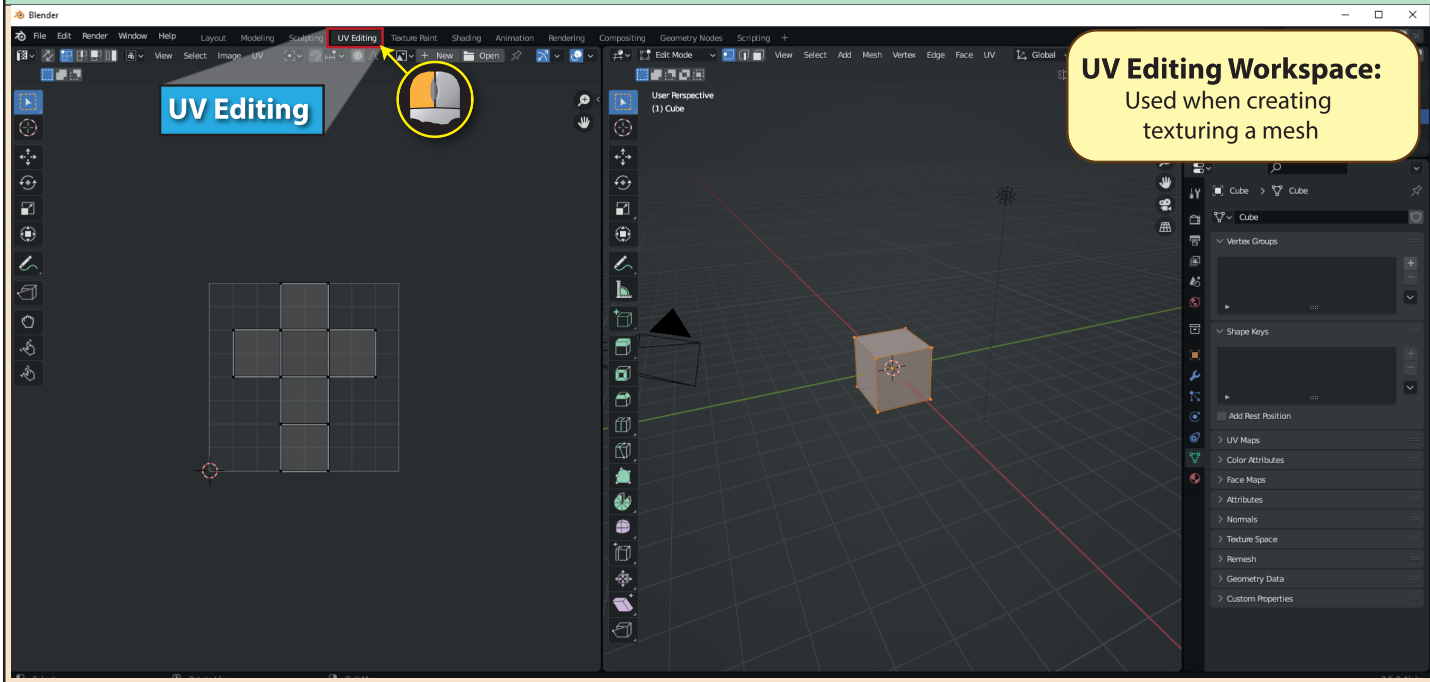


The second render option in the menu is ***Render Animation*** and, as the name suggests, this is used to render an animation video or a set of still frames showing the state of the animation at set time intervals (see below). More on animation in a later chapter.

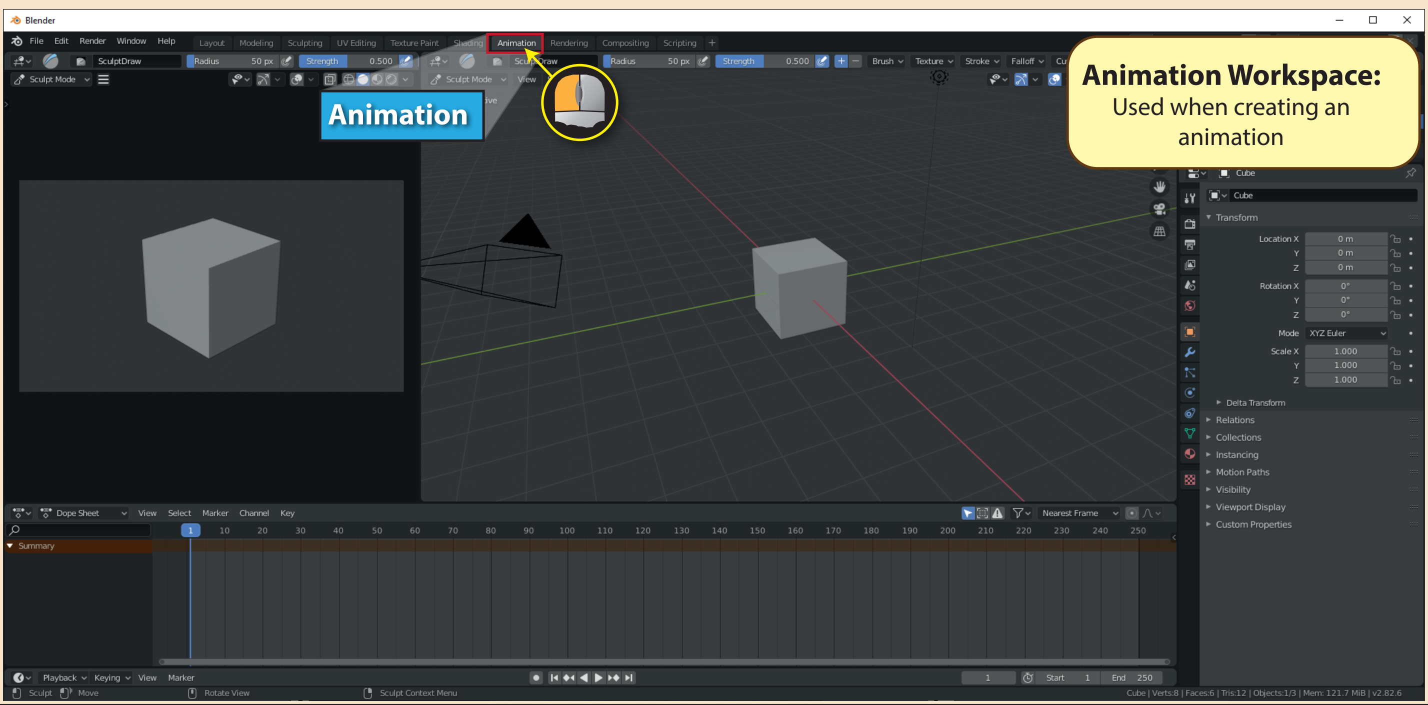
The second section of the top bar contains a set of tabbed pages known as **workspaces**. Each workspace has a different arrangement of editors. The different layouts are designed to be used at various stages throughout the creation of a scene.



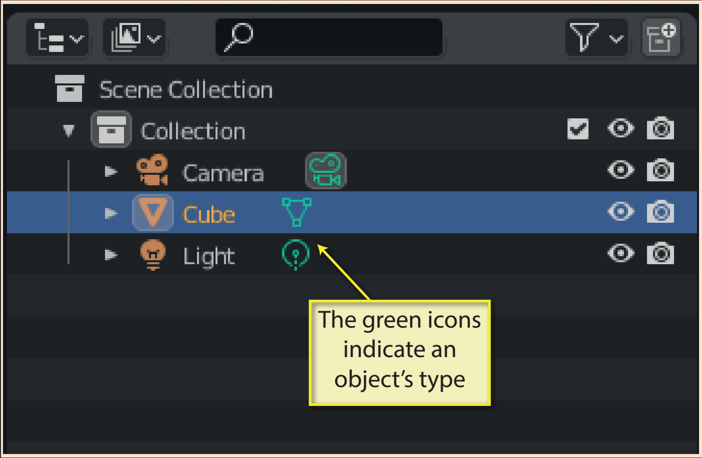
Below is the workspace created by clicking on the ***UV Editing*** layout option.



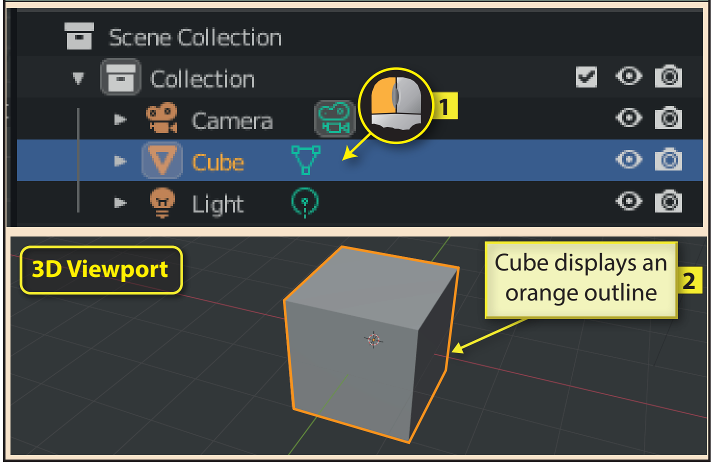
The ***Animation*** workspace option.



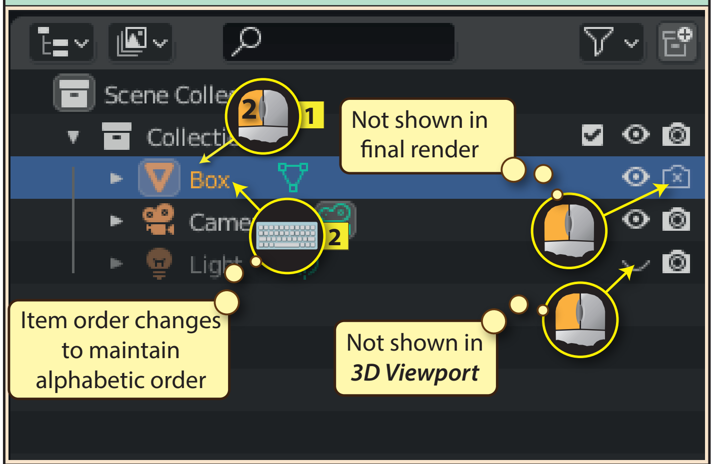
The **Outliner Editor** is near the top-right of the Blender window and lists all of the objects currently within the scene. When a new scene is created, there are three items already included: a ***camera***, ***cube*** and ***light*** and these are grouped together in **Collection**.



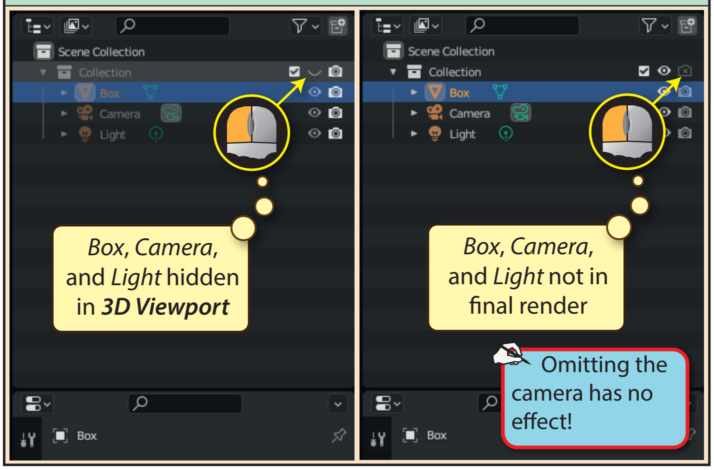
Clicking on the name of an item in the **Outliner** will select that item in the ***3D Viewport***. We can tell that an item is selected by the orange outline around it. Holding down the **Shift** key while clicking allows multiple items to be selected.



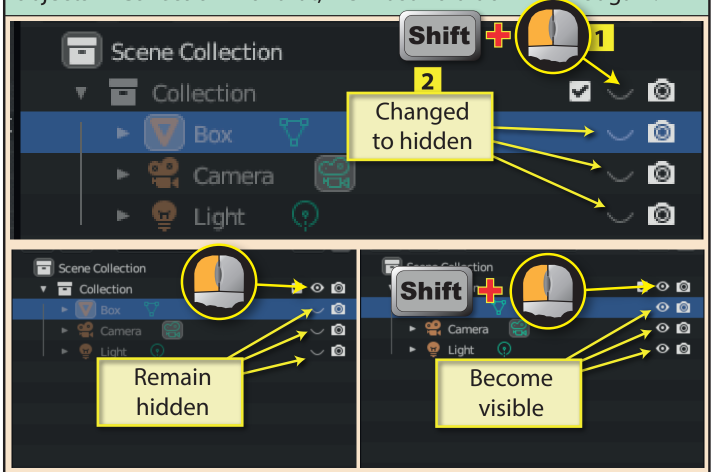
Within the **Outliner** we can change any item's name, make it visible/invisible in the ***3D Viewport,*** or have it included/excluded from the final render.



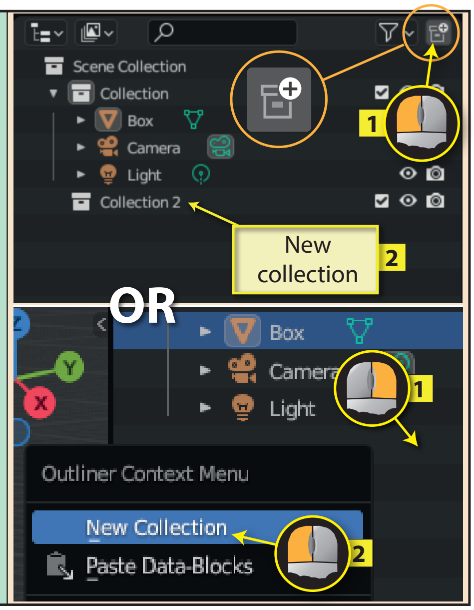
While clicking on the ***Eye*** or ***Camera*** icons of an object affects the that object, if we click on the same symbols to the right of ***Collection***, all of the objects listed under ***Collection*** are affected.



Holding down **Shift** when we click on ***Collection's*** *Eye* or *Camera* changes that setting on every object in the collection. Clicking without **Shift** will reset ***Collection's*** icon, but not the icon of the objects in ***Collection*** - for that, we must hold down **Shift** again.

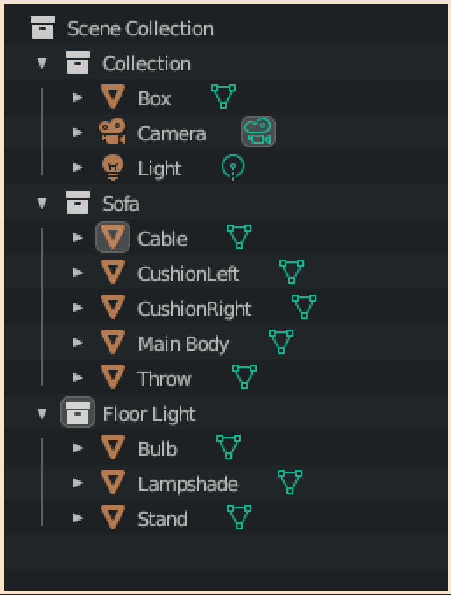


In a complex scene containing objects that are constructed from a number of meshes, it can be useful to create a collection for each object.
A new collection can be created by clicking on the *New* ***Collection*** icon (top-right) or by right clicking to produce a popup menu and selecting ***New Collection*** from its options.

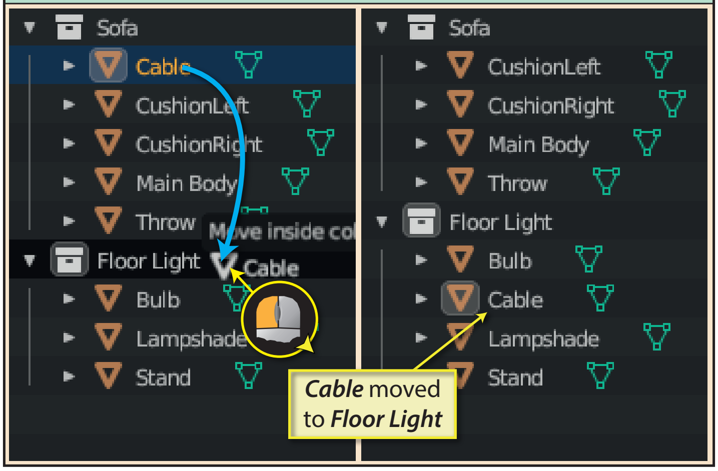


Typical use of collections is shown in the example given here where the elements that make up a sofa and a lamp stand have each been assigned an appropriately named collection.

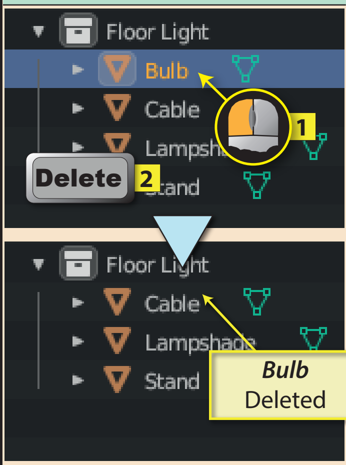
New meshes are added to the last collection to have been selected.



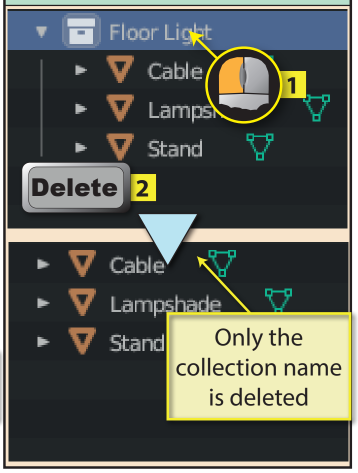
If we want to move an object to a different collection, all we need to do is drag the object into the required collection. For example, here ***Cable*** is dragged from ***Sofa*** to ***Floor Light***.



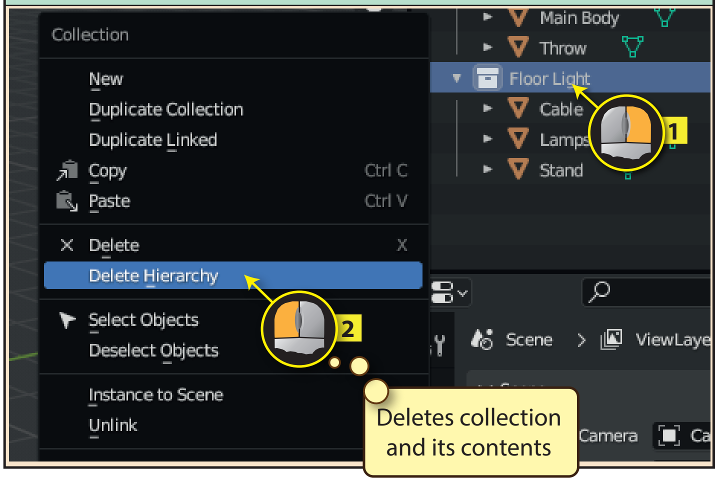
Selecting an object from anywhere in the ***Outliner*** and pressing **Delete** will delete that object.



Deleting a collection name removes the collection but not its contents. These items now belong to no collection.



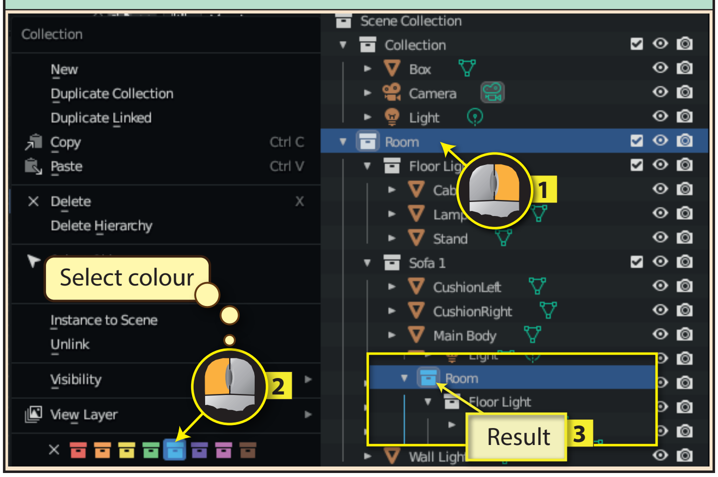
To delete a collection and its contents, we need to right-click on the collection name and select ***Delete Hierarchy*** from the popup menu.



We can even nest collections. For example, below we can see the sort of structure we might end up creating if our scene shows the contents of a room in a house.

A collection can be placed within another collection by dragging it to the required position.

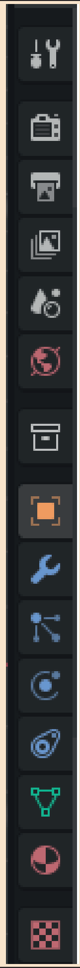
The icon associated with any given collection can be colour-coded. Select the collection, right-click and select the required colour in the popup menu.



The **Properties Editor** has by far the largest number of options of any editor panel.

A vertical list of icons on its left side act as a set of tabs with each one taking us to a related group of different options.

The icons shown here are those that appear when the Cube is selected.



Exactly which icons appear depends on the scene item currently selected.

In this frame we can see the set of icons associated with the **Light** object.

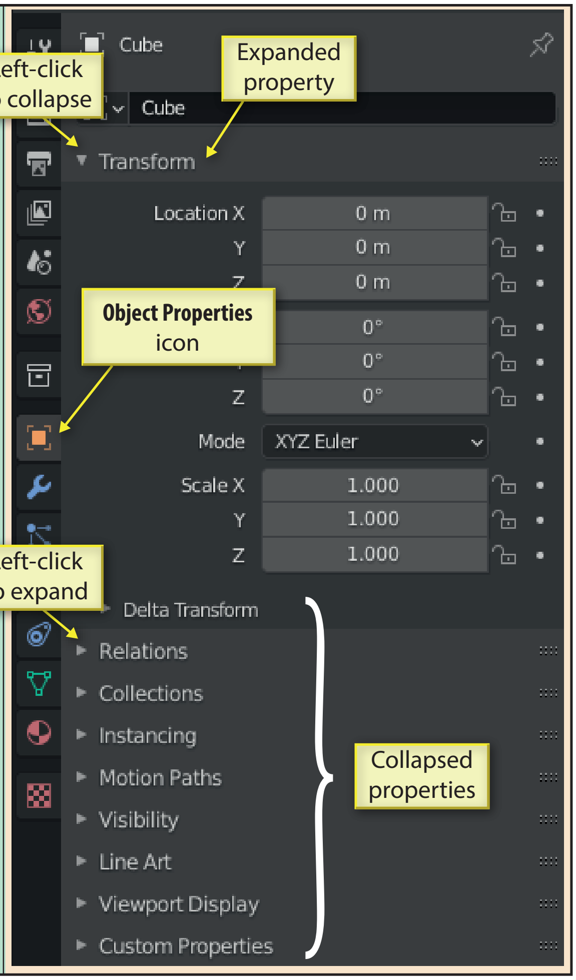
We'll look in detail at each option in a later section.

Clicking on one of these icons will display a set of properties on the right.

By default, the **Object Properties** icon will be selected.

Most property headings have several sub-properties.

Where a property title is collapsed, it displays a right-pointing triangle. Clicking on this will expand the property to show its set of subproperties.



Near the base of the Blender window is the **Timeline Editor**. This is only used when we are creating animations.

At this early stage in the learning process we may ignore the options shown here.

The lowest area in the Blender window is the **Status Bar**. This is used to display some of the options available to the user as a next operation.

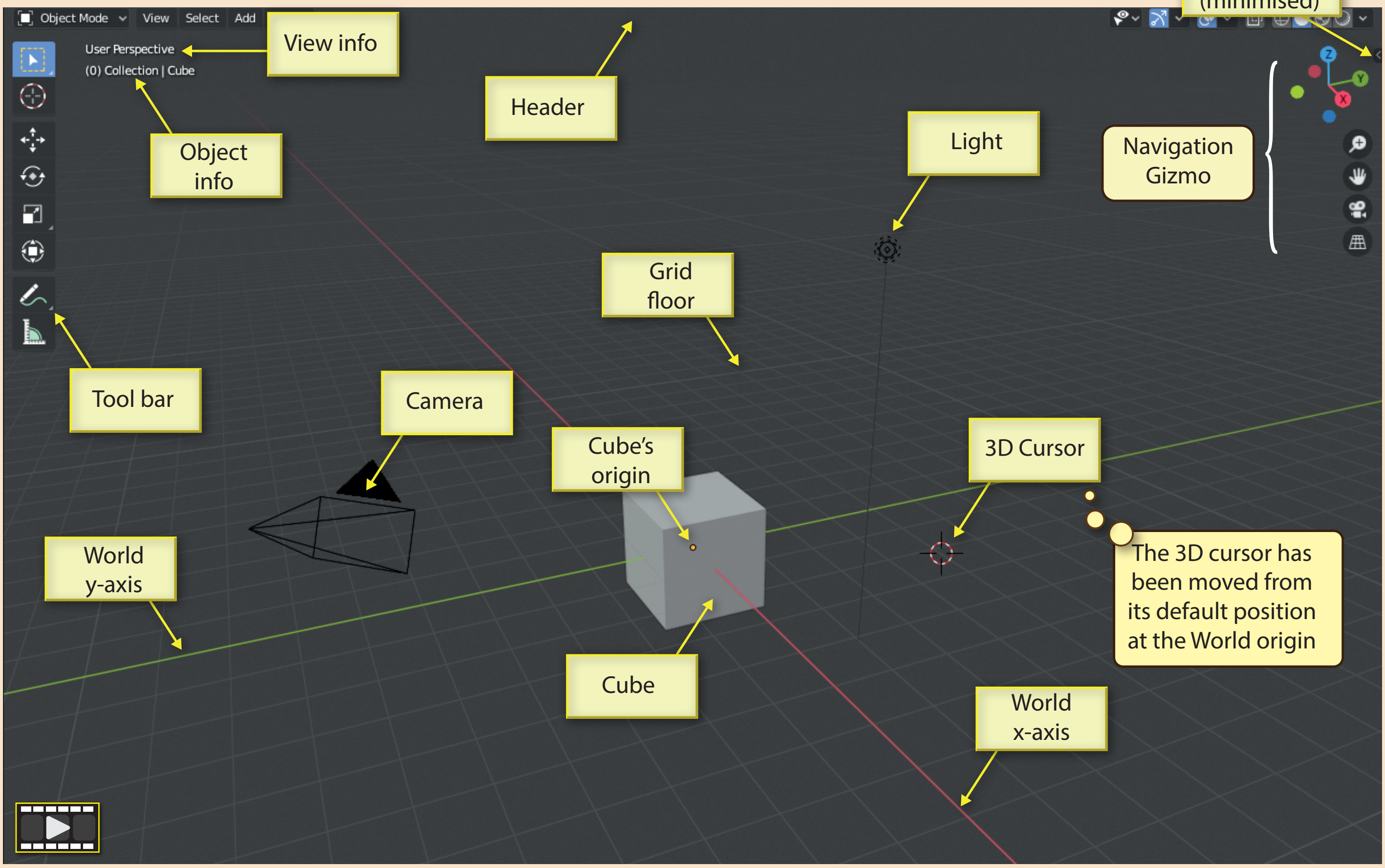
Shows that left-clicking on an object will select it

Dragging over multiple objects will select them all

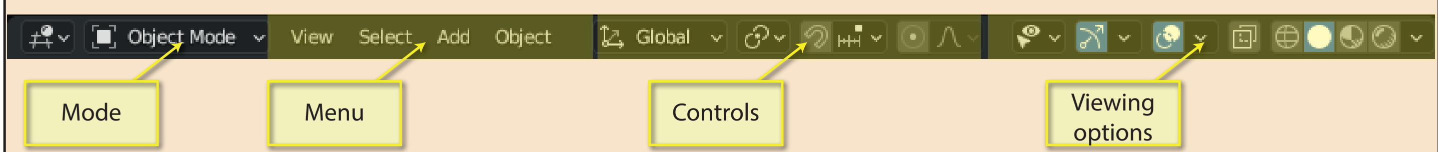
Dragging with the middle button will rotate the viewpoint

Right-clicking will produce a popup menu

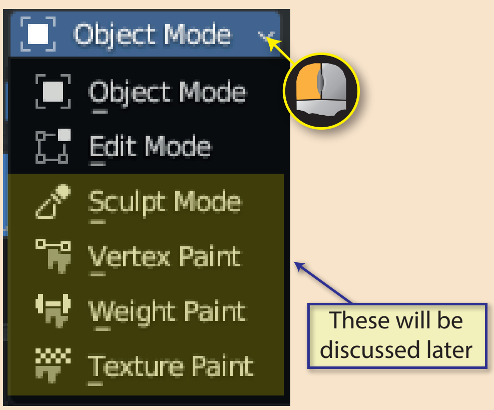
The default workspace is **Layout** and this contains the **3D Viewport Editor** which is our view into the 3D world. It is here that we will create our 3D model. There are several elements within this Editor as labelled below.



The **Heading area** of the *3D Viewport* contains four main areas. These show the current mode, a menu heading, edit controls and viewing options.



We'll be discussing each of these sections of the heading in more detail in a later chapter but for the moment we need only know that the **Mode** option dropdown has two entries of immediate importance: ***Object Mode*** and ***Edit Mode***.



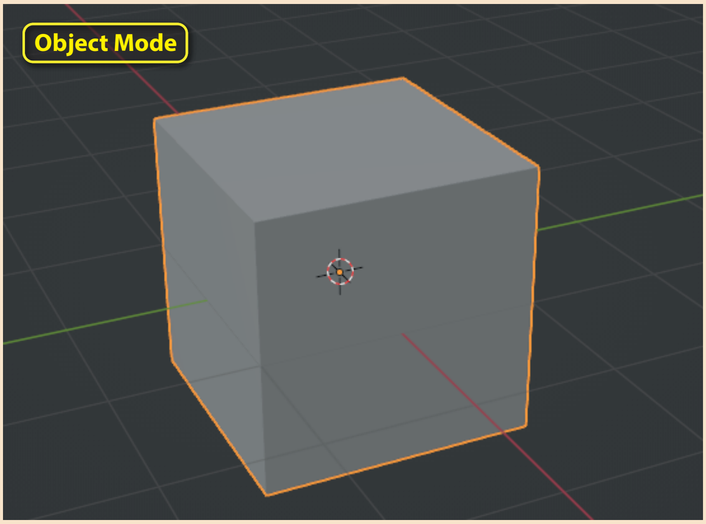
We work in **Object Mode** when we want to manipulate a mesh as a single object.

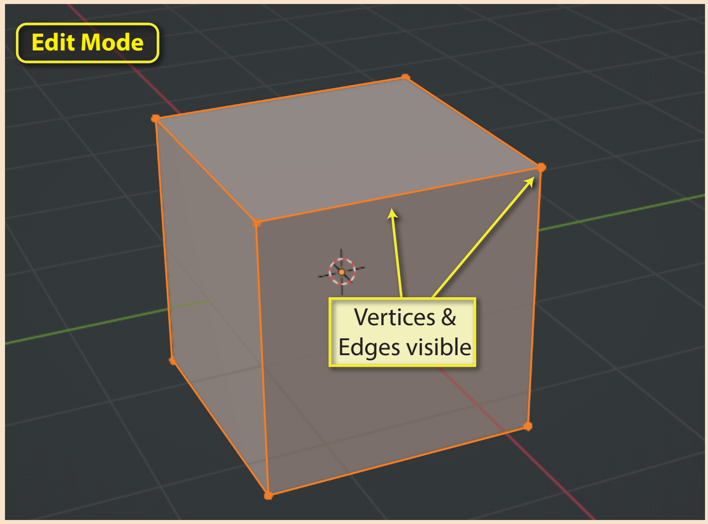
In **Edit Mode** we can manipulate the individual vertices, edges and faces within the selected mesh.

Although we can use the dropdown list in the ***Heading*** area to switch between ***Object*** and ***Edit*** modes, a quicker option is to press the **Tab** key which toggles between the two modes.

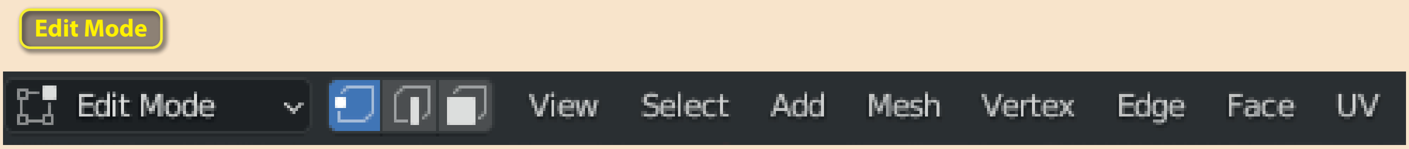
Switches between *Object* and *Edit* modes

Below, we can see how the image in the ***3D Viewport*** changes between ***Object Mode*** and ***Edit Mode***.

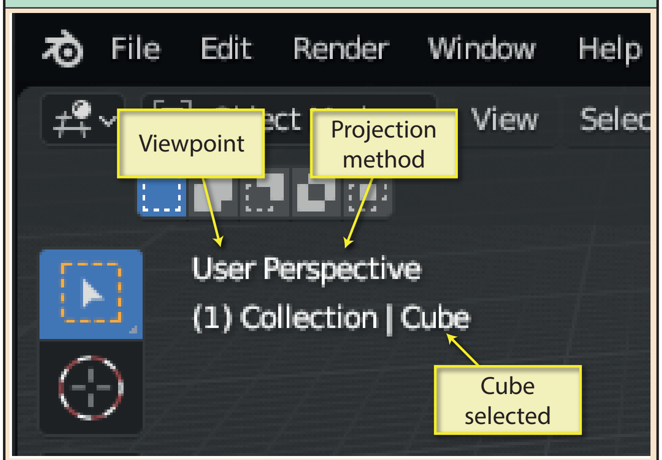




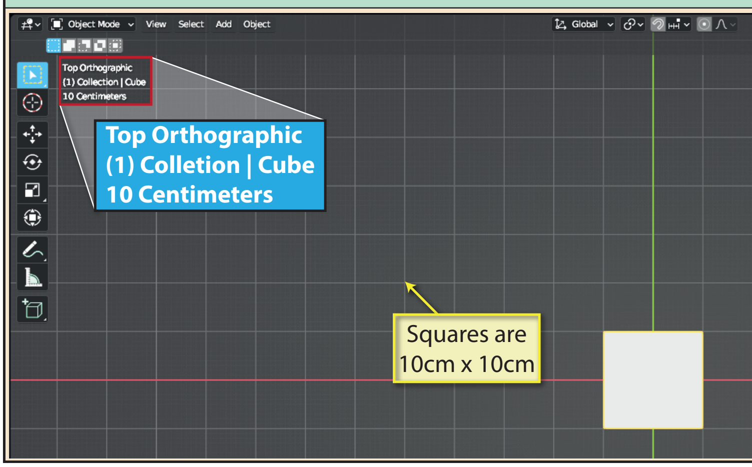
Notice that the menu options in the ***Heading*** area change depending on the mode we are using. Again, these will be discussed in detail in a later chapter.



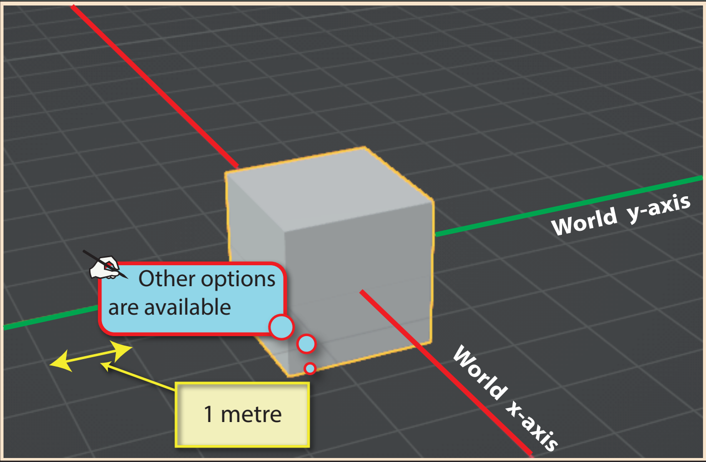
Below the heading is the main area of the ***3D Viewport***. Near the top-left of this area are two lines of text. The first line gives viewpoint and projection details; the second line, the currently selected object.



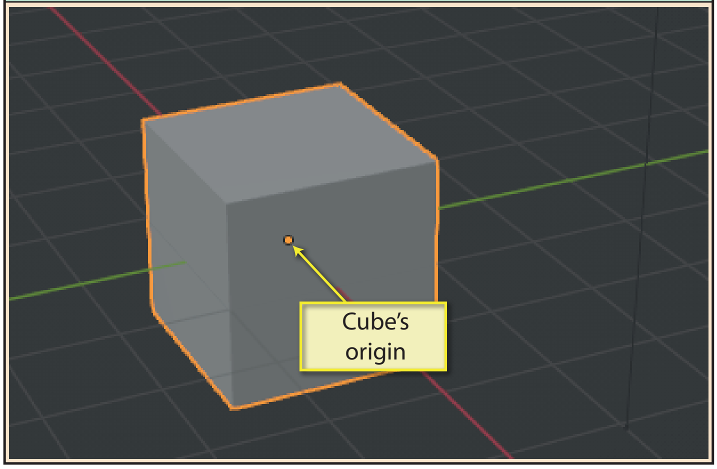
When we are in a named viewpoint such as *Front*, *Left* or *Top*, an extra line of text appears. This gives the size of the squares in the grid that is added to the background as an aid to measurement.



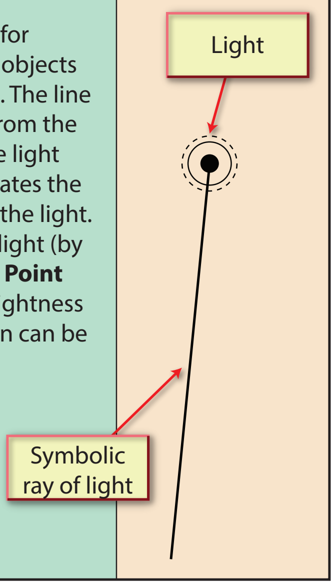
In the main area of the ***3D Viewport***, the **World** *x* and *y* axes are shown on the **grid floor**, but the *z*-axis is missing on start-up. The squares on the grid are **one Blender unit** in width and depth. By default, each Blender unit is equivalent to one metre.



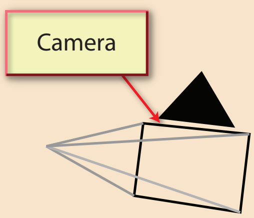
Blender automatically adds a ***Cube*** object to a new project. The ***Cube*** is one of the primitives available in Blender. The orange spot at the Cube's centre, is its **origin**. When we set coordinates for a mesh, it moves in such a way as to place its origin at that point.



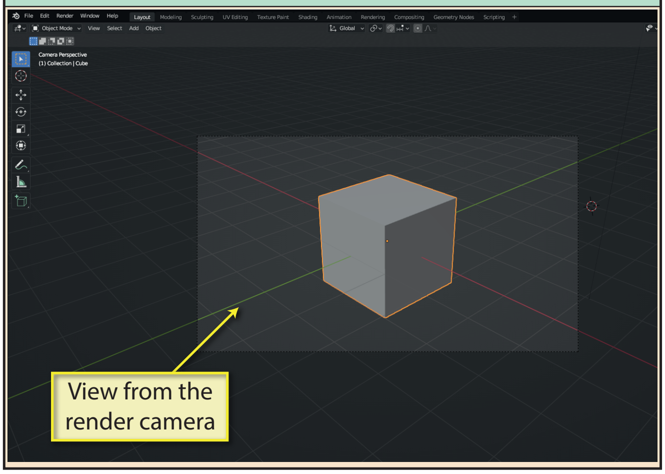
The light is responsible for lighting the objects in our scene. The line projecting from the centre of the light source indicates the direction of the light. The type of light (by default it's a **Point light**), its brightness and direction can be changed.



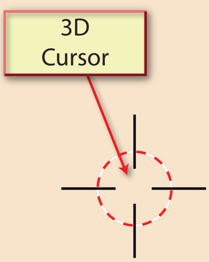
The view from the **camera** is what appears in the rendered image. We may think of the view we are seeing in the ***3D Viewport*** as coming from a second, invisible camera.



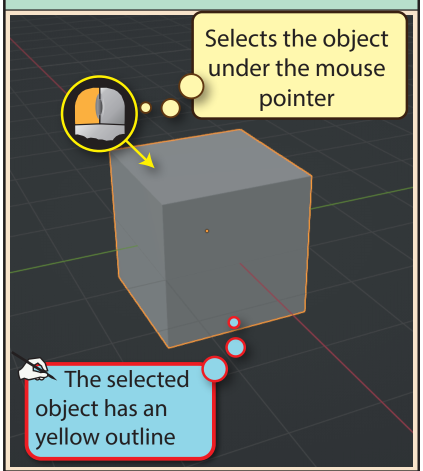
The view from the render camera is shown within the lighter coloured rectangle below. It is this area that appears in the final rendered image.



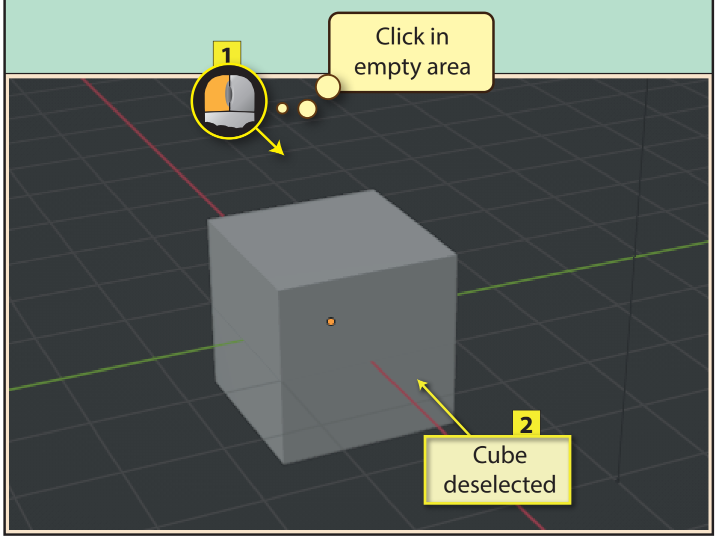
The **3D cursor**'s position determines where any new object is initially placed.
An object is placed so that its origin is positioned at the centre of the ***3D cursor***.



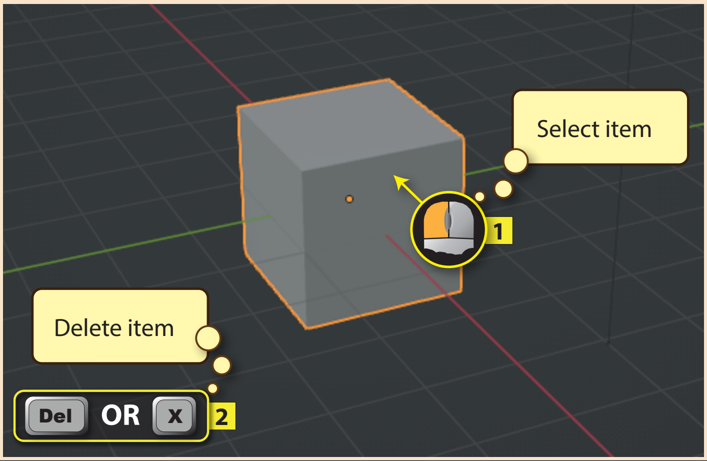
Although we can select a scene object by clicking on its name in the ***Outliner Editor***, a much simpler way is to left click on the object itself.



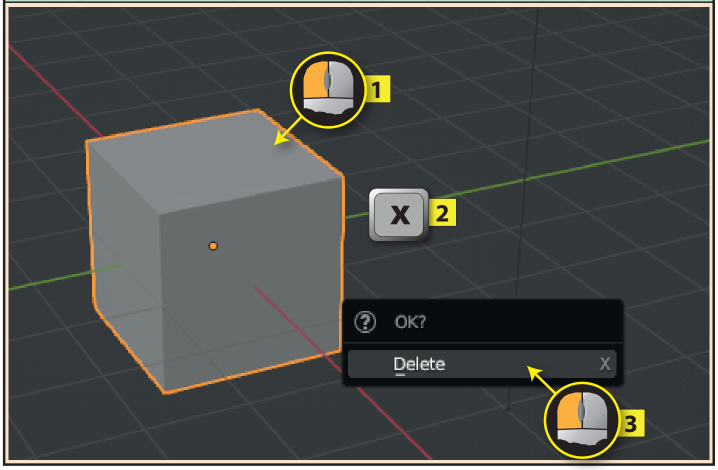
To deselect an item, all that is required is to left click in any empty area of the scene.



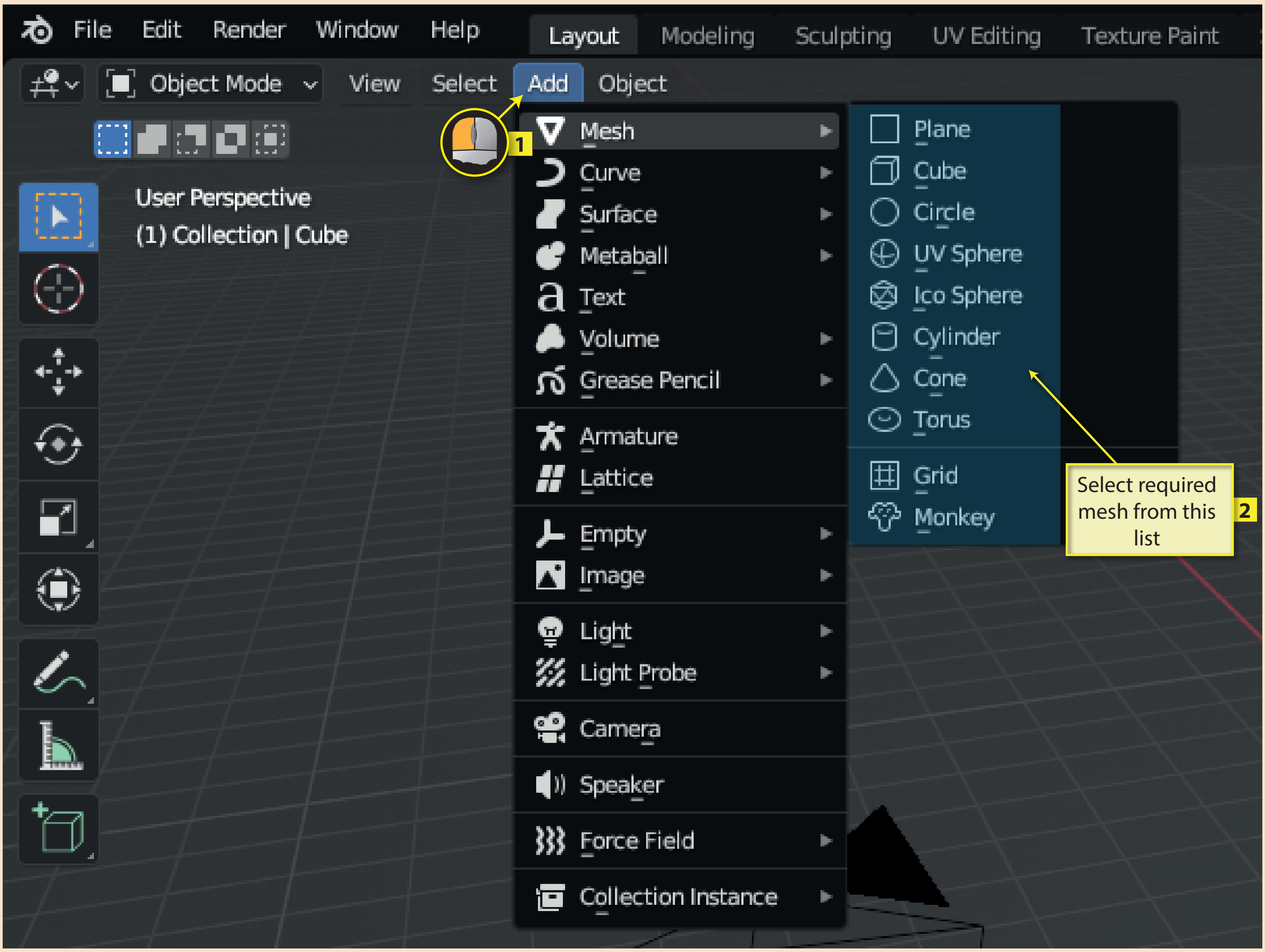
To delete an item from within the ***3D Viewport***, we must first select it and then press either the **Delete** key or the **X** key.



While the **Delete** key deletes the item immediately, if the **X** key option has been used, then Blender will require confirmation of the action before removing the object.

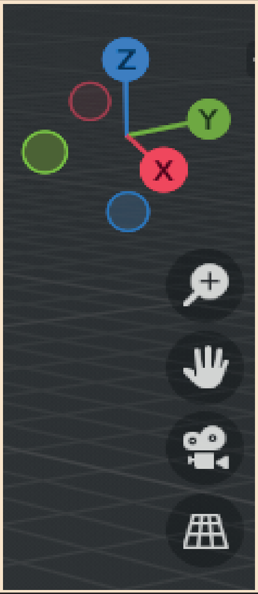


We can add new meshes to our scene by selecting the **Add|Mesh** option from the ***3D Viewport's*** own main menu.

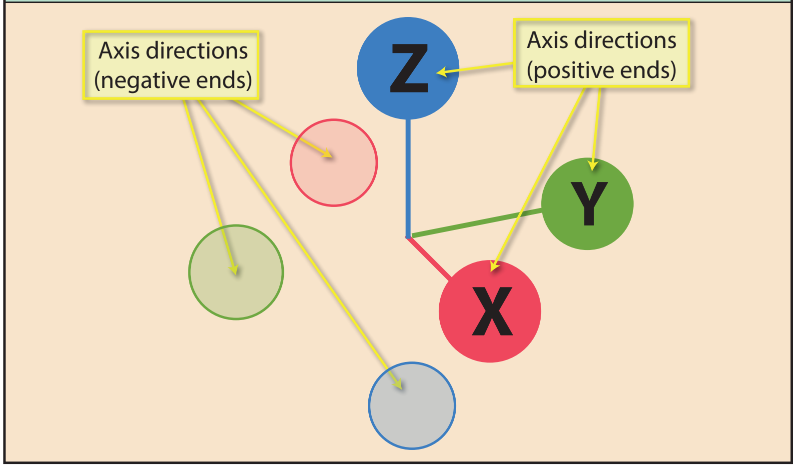


Navigation

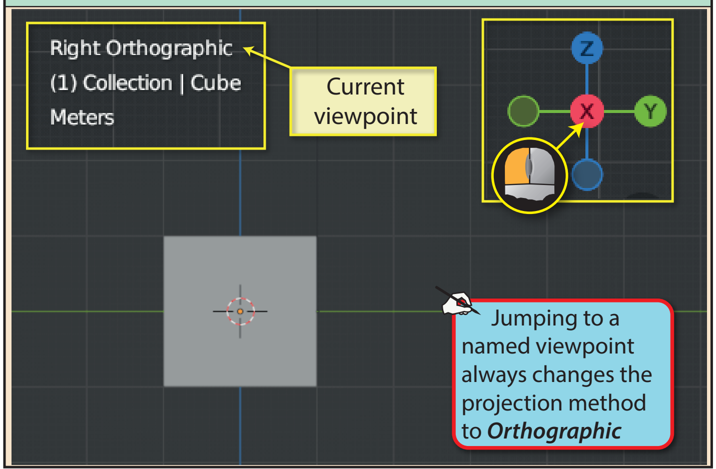
The **Navigation gizmo** - in the top right of the ***3D Viewport*** - contains several elements which we can use to perform various tasks such as change viewpoint, zoom in and out, and switch between perspective and orthographic view.



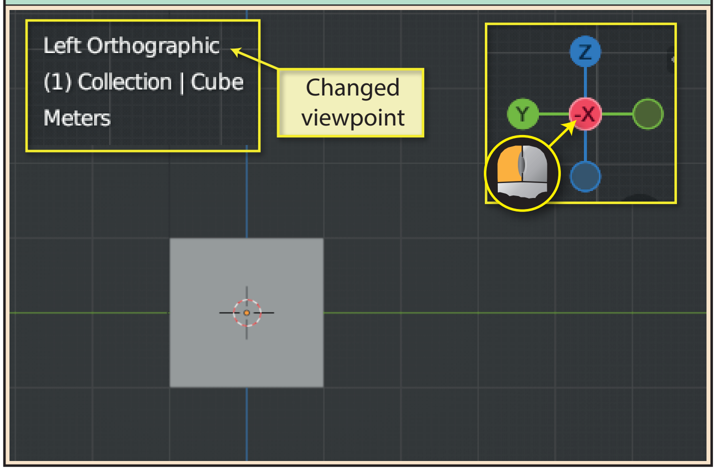
The first of these elements shows the orientation of the **World Axes** relative to the current viewpoint. Labels are displayed in the positive end circles and in the duller, negative ends only when the mouse pointer moves over them.



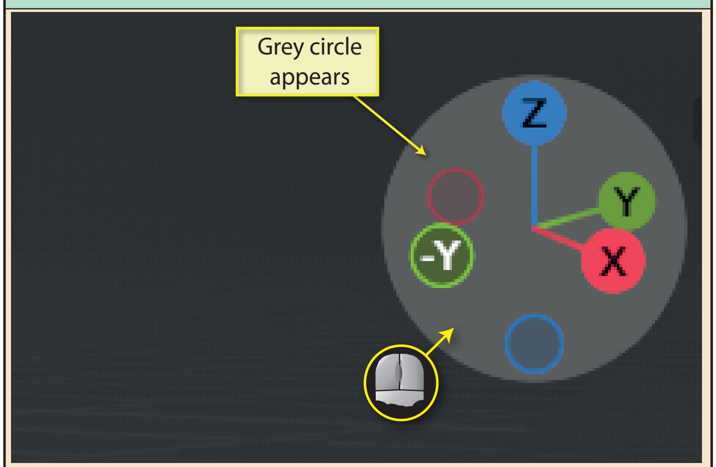
Clicking on one of the circles will jump the viewpoint in the ***3D viewport*** to a named direction (**Front**, **Back**, **Left**, **Right**, **Top**, or **Bottom**). The direction depends on which circle is clicked. The current viewpoint (and other details) appear at the top-left.



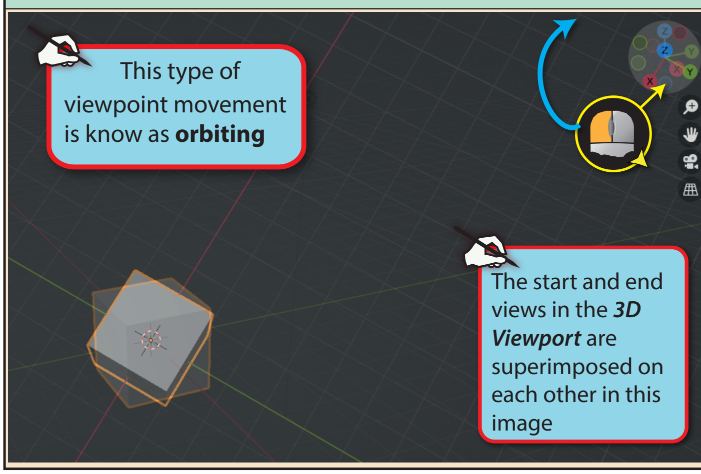
Clicking on the same circle as before (**X**) changes the view by 180°. In this case, to the ***Left view***. Of course, in the case of a cube, there's not much difference!



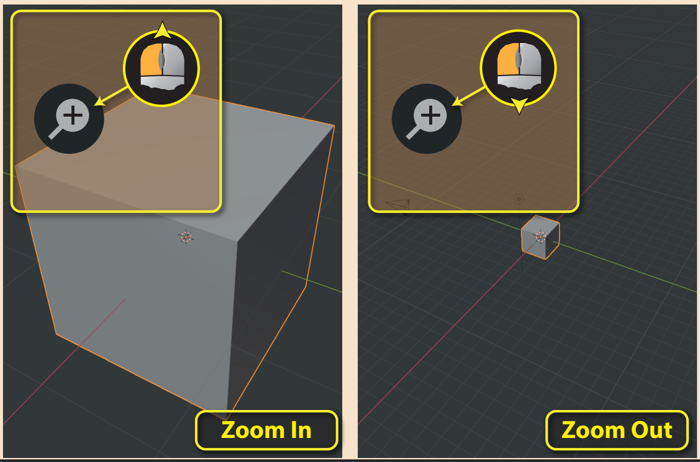
If the mouse pointer is moved into the dark area between the coloured circles, a new, large grey circle will appear.



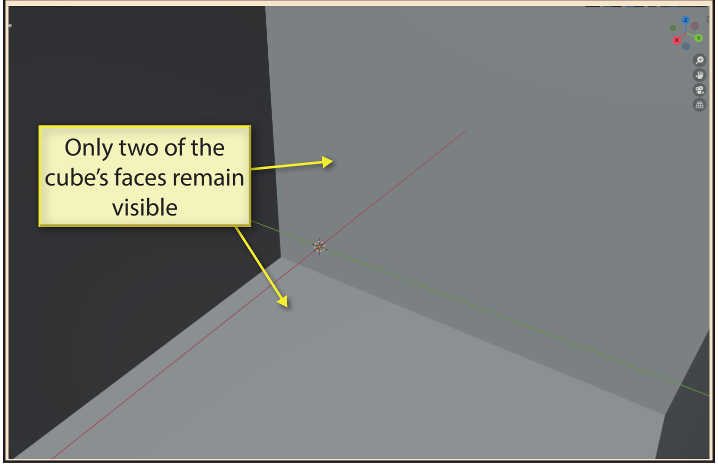
Starting a mouse drag within the grey circle allows freestyle circular movement of the viewpoint about the centre of the screen.



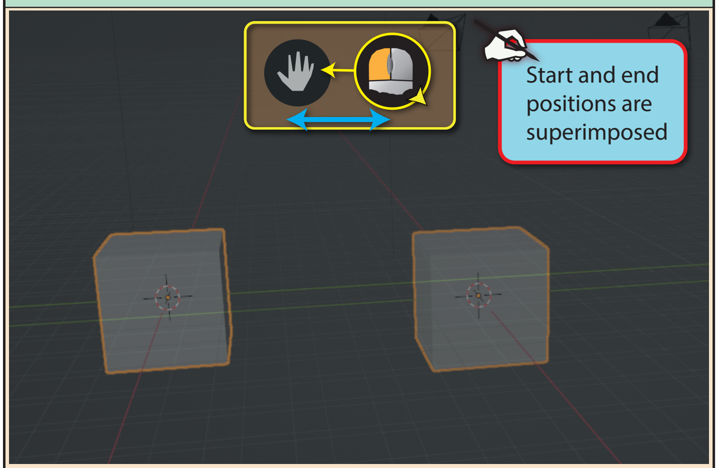
The **Zoom icon** is the next element of the ***Navigation Gizmo***. Dragging the mouse up in this area will zoom in on the scene. Dragging down will zoom out.



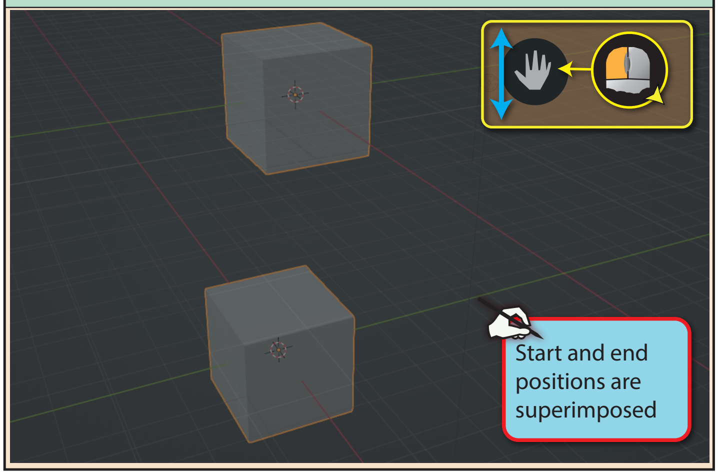
However, be aware that if we zoom in too far, faces nearest the viewing camera may disappear from view. In the example below most of the cube's faces are no longer visible within the ***3D Viewport*** and we have moved inside the Cube.



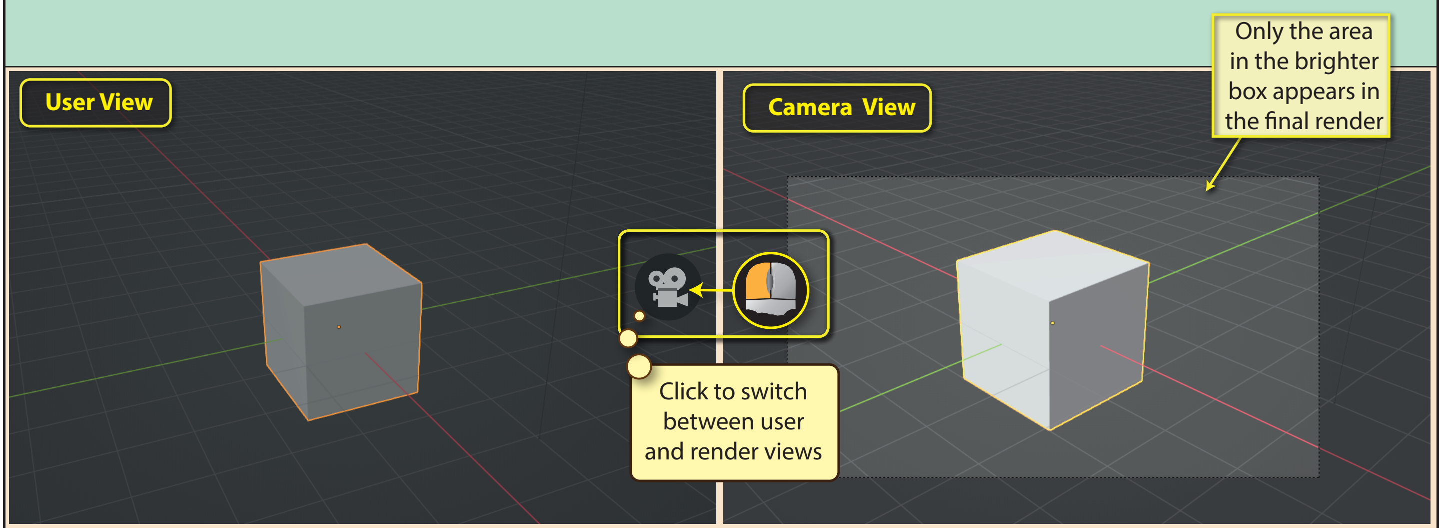
By first placing the mouse pointer over the **Move icon** then dragging in the left or right direction allows us to move our viewpoint to the side.



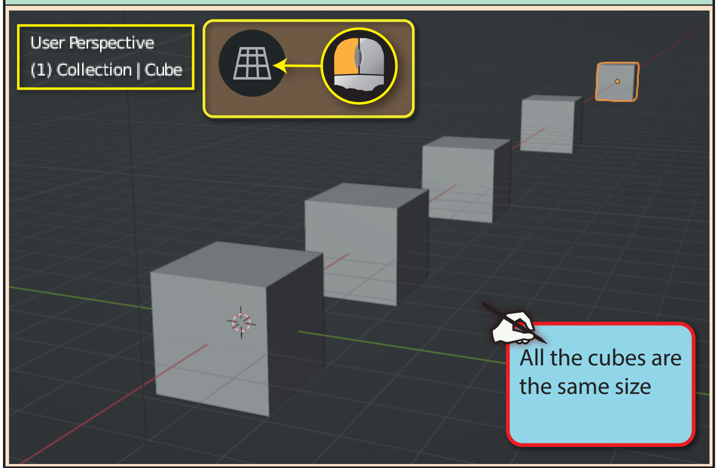
Dragging vertically moves our viewpoint up or down.



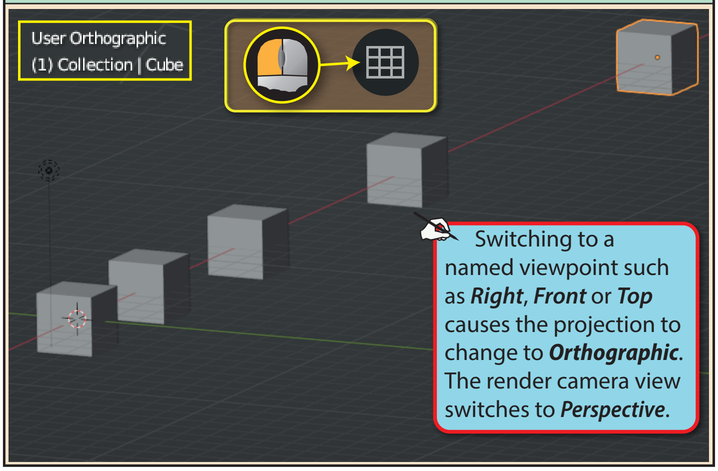
The **Camera View icon** allows us to toggle between the view from the camera object (which is used when rendering the final image) and the normal (user's) viewpoint. To activate this option, click on the icon.



The **Projection icon** allows us to toggle between **perspective view** (where items further away appear smaller)...



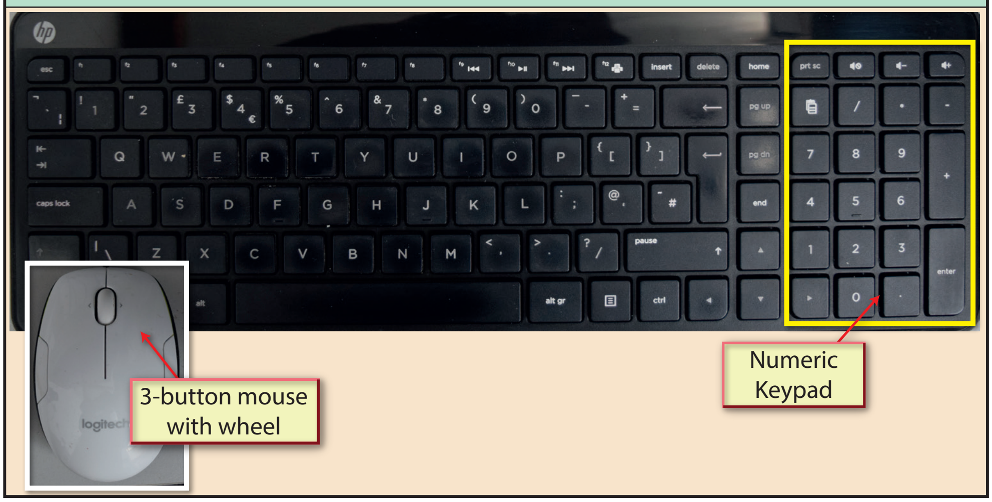
...and **orthographic view** where equally sized items are of identical size irrespective of their distance (although your brain may be fooled into thinking ones further away are larger).



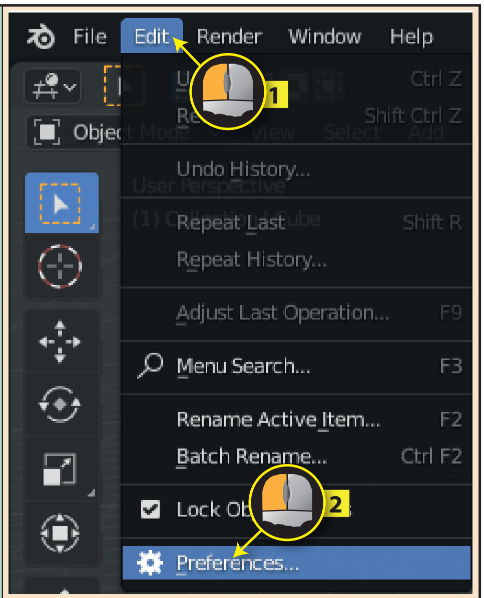
We've already seen that the ***Navigation Gizmo*** allows us to easily move to a new viewpoint as well as zoom and switch between perspective and orthographic views.

But Blender offers multiple ways of performing these tasks, many of which are quicker and simpler than using the ***Navigation Gizmo***.

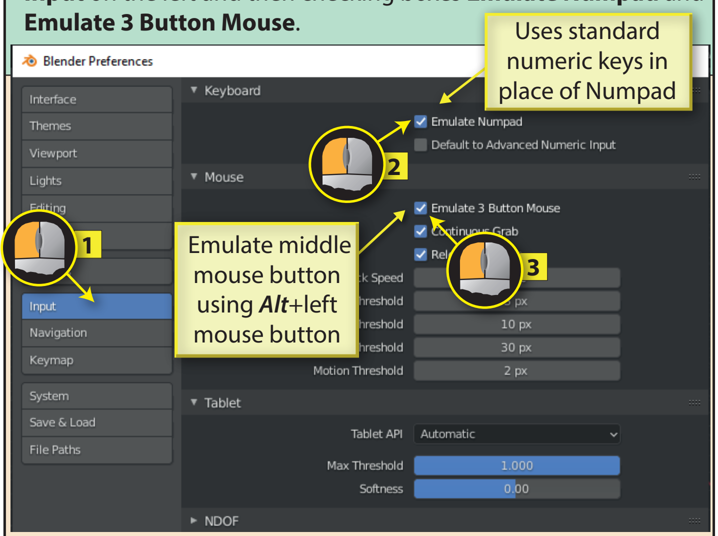
It should be pointed out, before we start, that life will be a lot easier if we are using a full-sized keyboard which includes a separate numeric keypad section (**numpad**). Also having a three-button mouse (the middle button being a scrollwheel) would be useful.



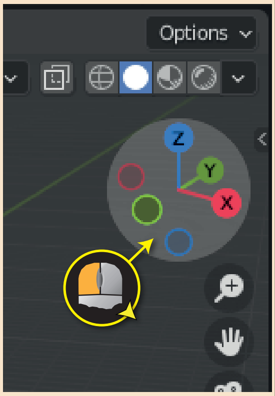
We can emulate the existence of the numpad and three button mouse if necessary by first selecting ***Edit> Preference*** from the main menu...



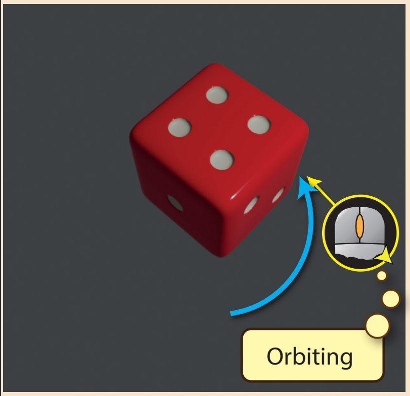
...to produce the *Blender Preferences* window. then clicking on **Input** on the left and then checking boxes **Emulate Numpad** and **Emulate 3 Button Mouse**.



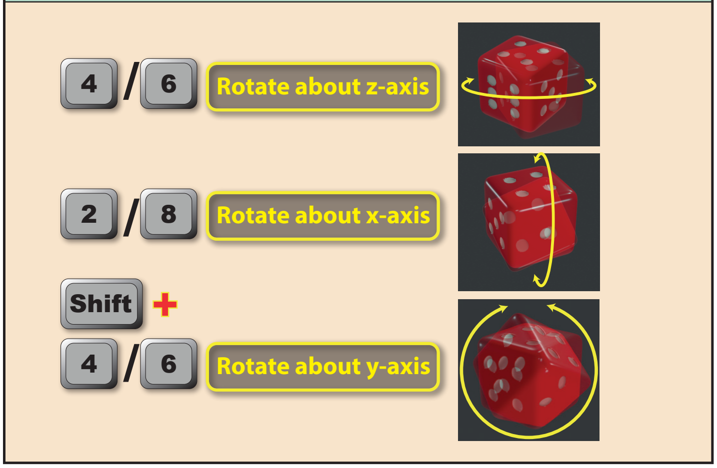
We've already seen how the **Navigation Gizmo** can be used to change our viewpoint by dragging the mouse within the grey circle.



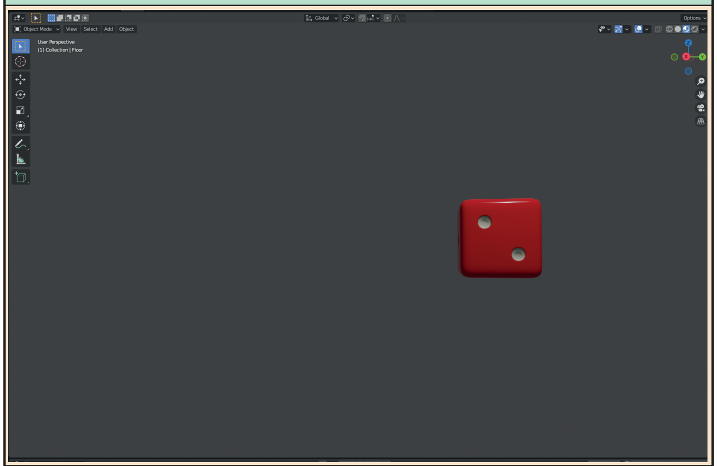
But the simplest way to achieve the same effect is to hold down the **middle mouse button** (or **Alt**+ left mouse button if you are using emulation) and drag the mouse anywhere in the ***3D Viewport***.



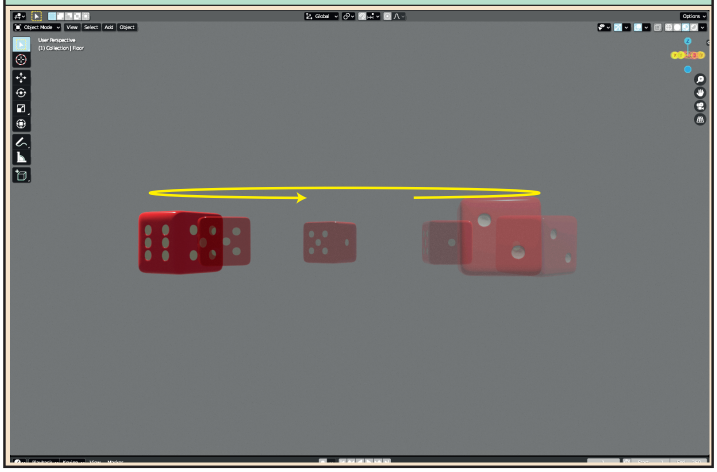
Another way of orbiting is to use keys on the numpad. Pressing **4** or **6** rotates the view about the user's *z*-axis (in opposite directions) while **2** or **8** rotate it about the *x*-axis. Also, **Shift + 4** or **6** rotates about the *y*-axis. Each keypress rotates the view by 15°.



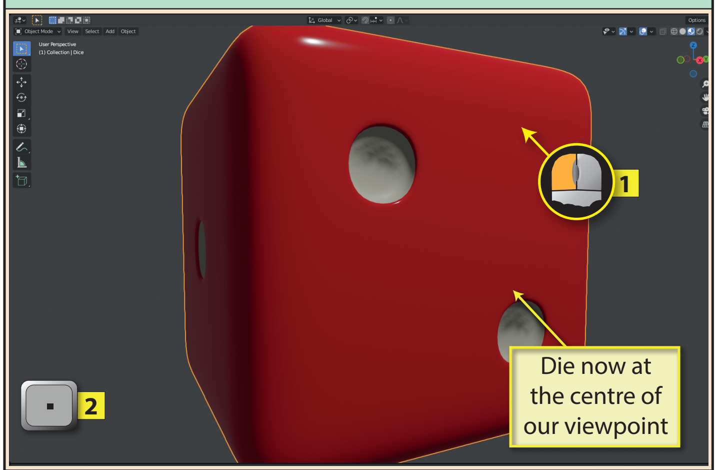
Orbiting is always around the centre of the screen. This can cause problems if the object we want to orbit is not at the screen's centre. For example, if we start in the position shown below...



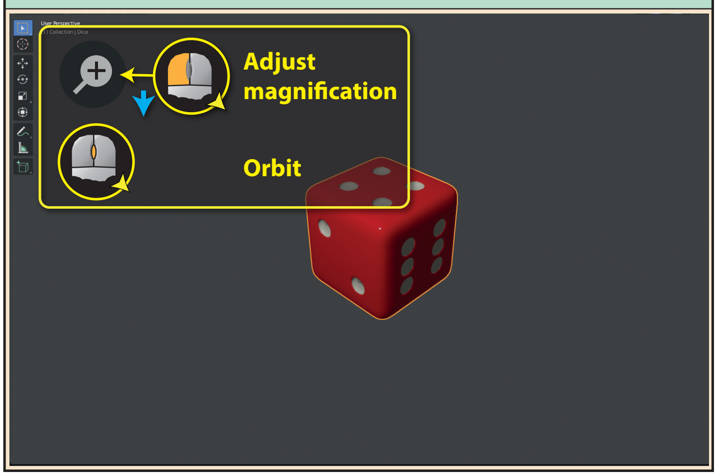
... and orbit using the middle mouse button, we get the rotation shown below with the die apparently circling the centre of the screen (remember, it's our viewpoint that is changing, not the position of the die).



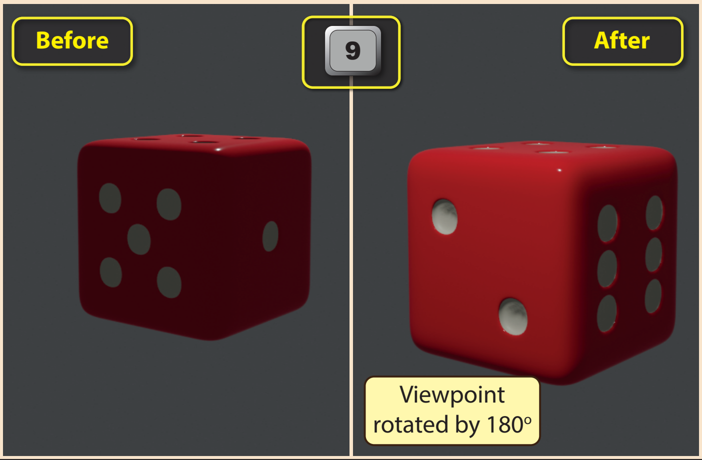
To avoid this effect, we start by selecting the object we want to orbit (in this case, the die) and then press the period (full stop) key on the numpad. This moves our viewpoint so that the die is centred on the screen (and usually zooms in on it too).



We can zoom out if necessary (drag on the ***Magnify icon***) and then orbit in the usual way (dragging on the middle mouse button).

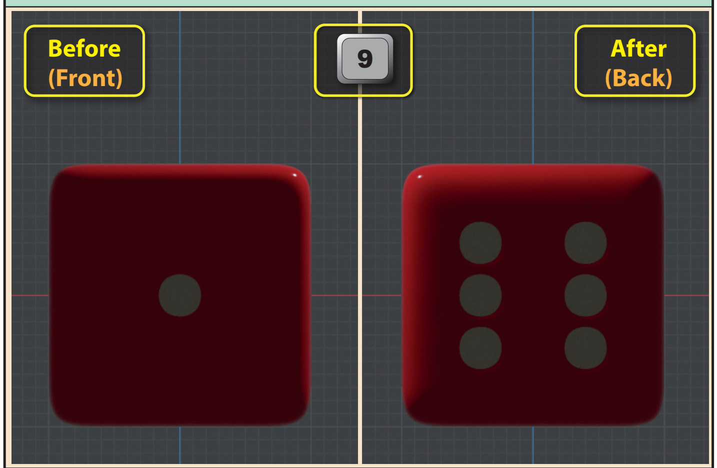


The numpad's **9** key rotates the view by 180° about the viewer's *z*-axis.

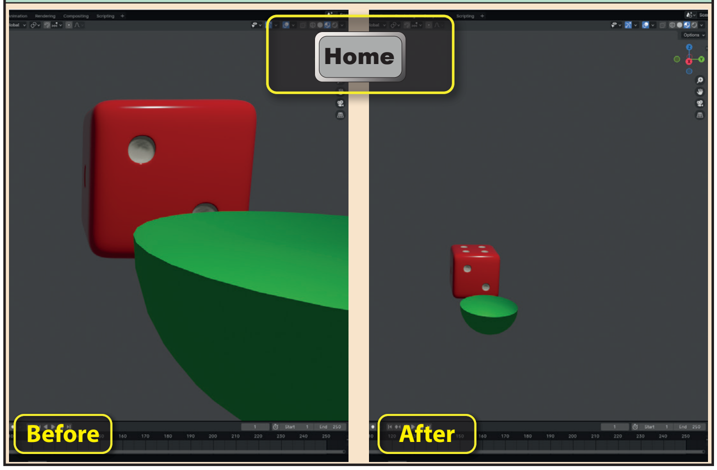


Named viewpoints such as ***Front***, ***Left*** and ***Top*** can be accessed by pressing the appropriate key on the numpad or in combination with the **Ctrl** key (as shown here).

Since pressing **9** rotates the viewpoint by 180°, it is a useful alternative way of toggling between named viewpoints such as ***Front*** and ***Back***.



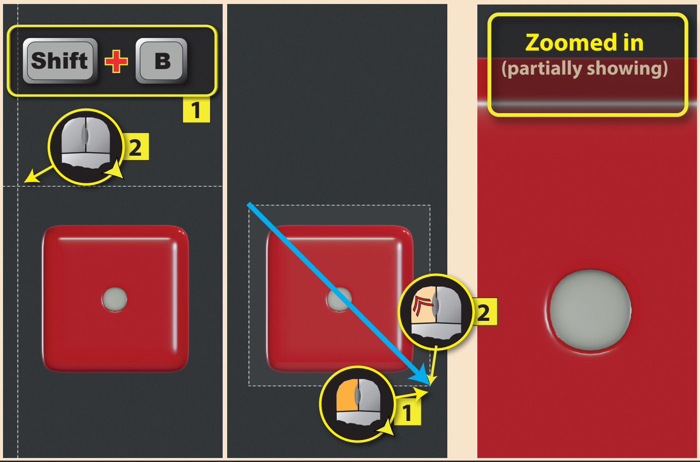
We can jump to a view of the whole scene by pressing the **Home** key.
(In the "before" and "after" images below, only the right side of the ***3D Viewport*** is shown.)



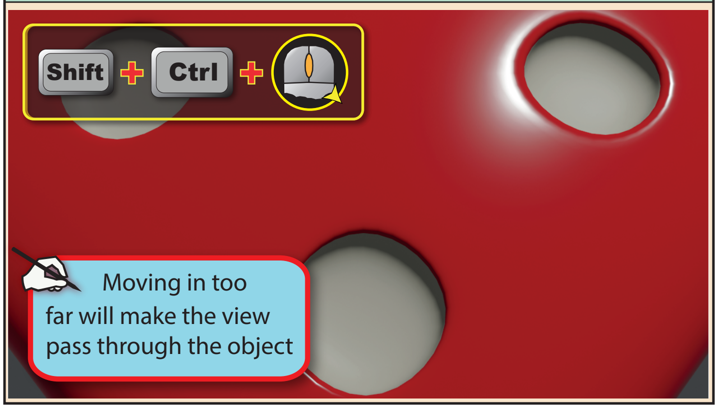
The other icons in the **Navigation Gizmo** also have numpad or mouse movement equivalents.

	Zoom	Panning	Camera	Projection
Gizmo				
Mouse		Shift + Pan left/right up/down		
Numpad	+ Zoom in - Zoom out	Ctrl + 4 Left Ctrl + 6 Right Ctrl + 8 Up Ctrl + 2 Down	0 Toggle View/ Render Camera	5 Toggle Perspective/ Orthographic

A little used zoom option allows us to draw a box around the area to be zoomed into. Pressing **Shift+B** sets up the box drawing, dragging and releasing the left mouse button defines the size of the box and Blender then zooms into that area.



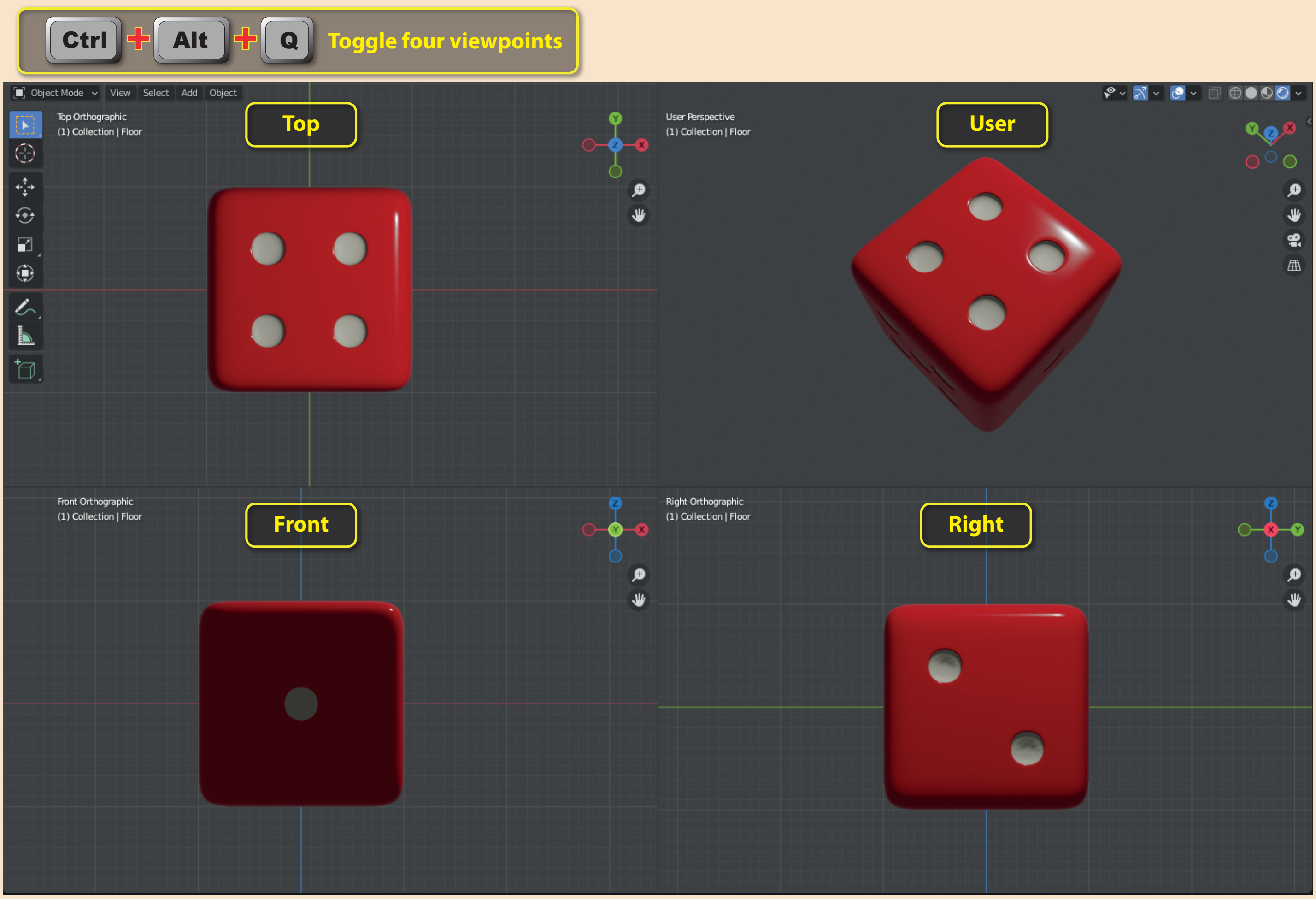
All methods of zoom that we've looked at so far limit the amount of zoom available. But a final zoom option in Blender is the equivalent of moving the viewpoint camera itself closer or further away from the subject. This method (known as **dolly zoom**) requires us to hold down **Shift+Ctrl** while dragging with the middle mouse button.



Later, when we start to create and edit scenes, it can be helpful to be able to see an object from several viewpoints at the same time. We can achieve this by pressing **Ctrl+Alt+Q** which splits the viewport into four equally sized view windows. Three of the windows show the named viewpoints, ***Top***, ***Right*** and ***Front*** while the fourth retains the user's current viewpoint.

Each of these view windows acts as an independent viewpoint. When we move the mouse pointer into a specific window we can adjust that view in any way we wish such as zooming, panning or switching to another named viewpoint.

To merge the four views back to a single image in the viewport, we need only press the same key combination, **Ctrl+Alt+Q**.

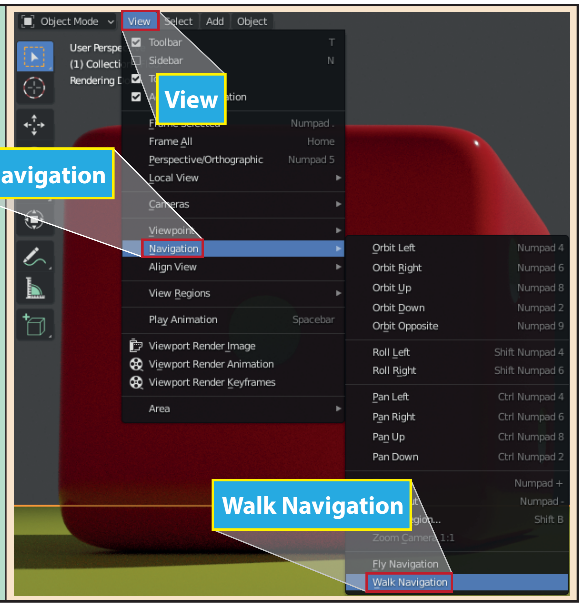


In large, complex scenes such as a cityscape, the methods we have seen so far for changing the viewpoint may not work as well as we would like.

An alternative approach is to allow the viewpoint changes to use the same techniques as we see in a typical first-person shooter (FPS) game.

This approach allows us to "walk" through our scene with any zooming, panning or orbiting happening relative to our current position within the scene.

To access **Walk Navigation** we need to select ***View>Navigation>Walk Navigation*** from the main menu.



Once in "walk" mode, moving the mouse pointer (no button presses) allows us to look around while the keys **W, S, A, D, E, Q** and the arrow keys control navigation.

E and **Q** only work when gravity is off (see below).

Pressing **Esc** will exit "walk" mode.

Look around

▲ / W Move forward

▼ / S Move backward

◀ / A Sidestep left

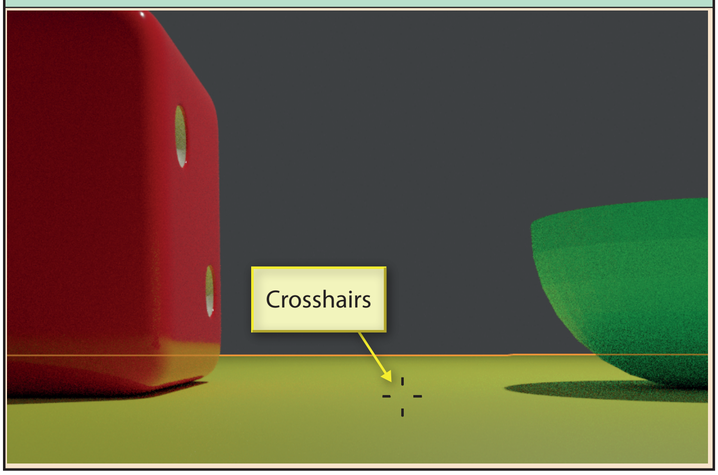
▶ / D Sidestep right

E Move up

Q Move down

When we switch to walk mode we'll also notice that the mouse cursor changes to the typical crosshairs of a FPS game.

Pressing the left mouse button returns us to normal navigation.



Holding down the **Shift** key while pressing one of the navigation buttons will speed up the movement.
Holding down **Alt** while navigating will slow down movement.
To permamently change the speed, scroll the mouse wheel or use the numpad's + or - keys.

Temporarily Increase Speed

Shift + { W, S, A, D, E, Q }

Temporarily Decrease Speed

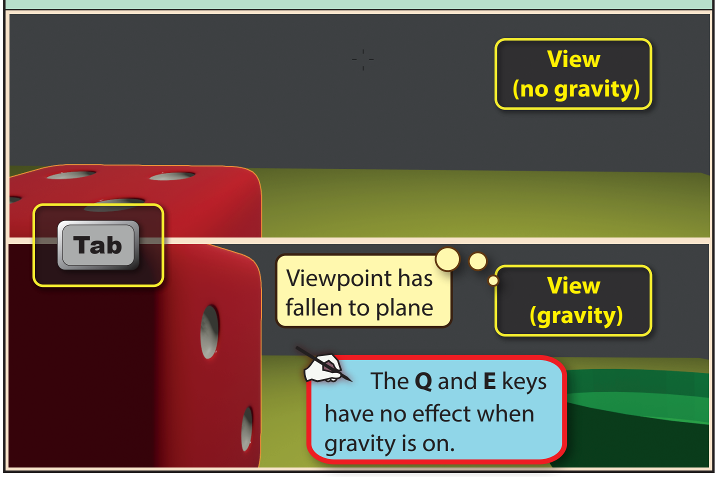
Alt + { W, S, A, D, E, Q }

Permanent Speed Change

Increase / +

Decrease / -

While in "walk" mode, pressing the **Tab** key will introduce the effect of gravity causing the user's view to fall until a solid surface is encountered. Pressing **Tab** again will remove the gravity effect.



Viewing a Scene

1

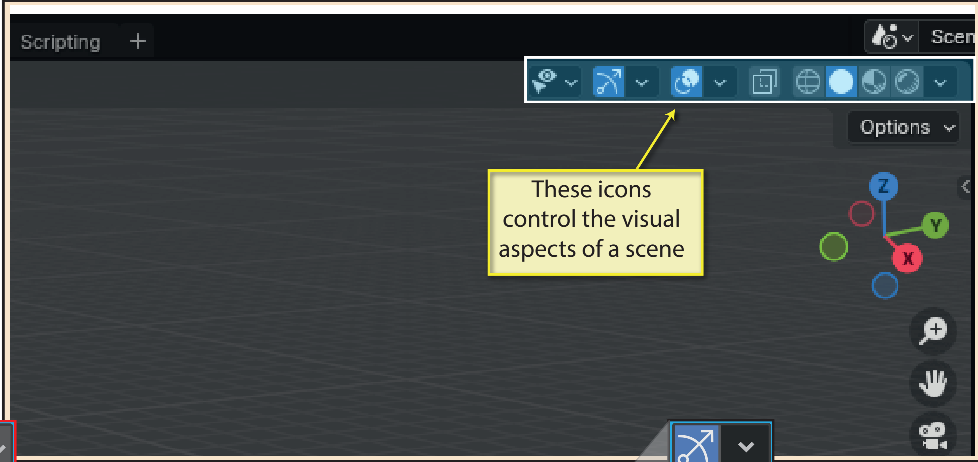
2

When working in the *3D Viewport* there are several options available to us to control what elements are visible or selectable within our scene.

We can also control the colour of objects and the lighting of our scene.

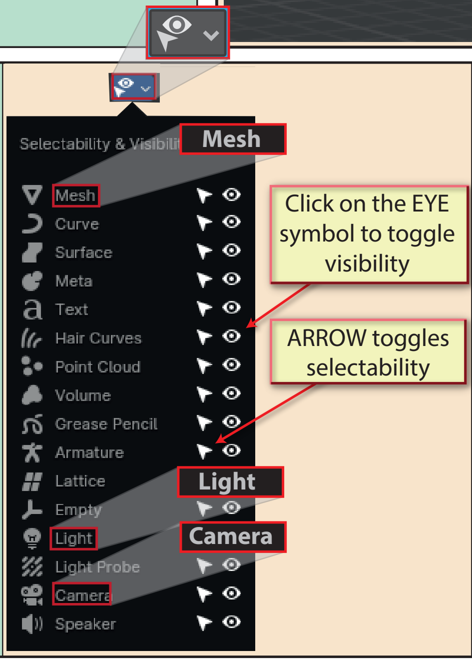
These colours and lighting effects can be set to be applied only while we are working on our scene or to be applied to the final render.

At the right of the **Header area** there are a set of icons that are resposible for controlling what and how we see the elements in our scene.



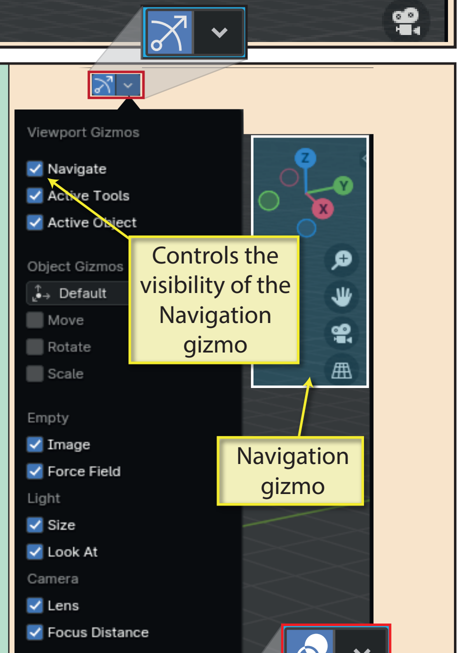
The first of these icons, is the **Selectability and Visibility** settings.

Clicking on its dropdown button displays a set of options which we can use to hide a whole class of objects with a single click or make that group of objects unselectable.



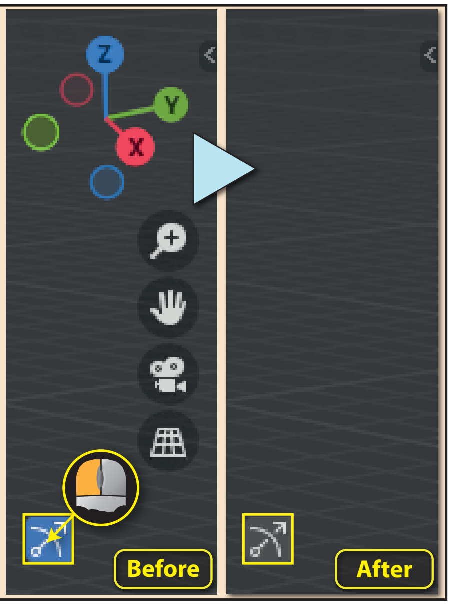
The second icon is named the **Gizmo Visibility** icon and affects the visibility of various controls that appear within the *3D Viewport*.

At this point we need only know that the first checkbox controls the visibilty of the ***Navigation Gizmo*** icons.



Notice that the **Gizmo Visibility** icon has a blue background.

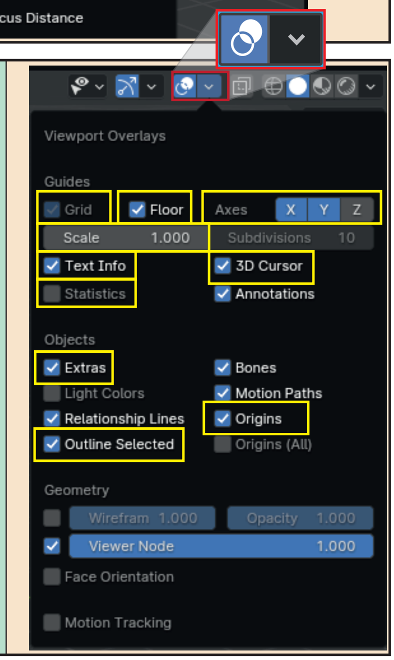
This indicates that the options selected within its panel are active.
If we click on the icon itself, we can switch off all the options it contains as indicated by its now grey background.



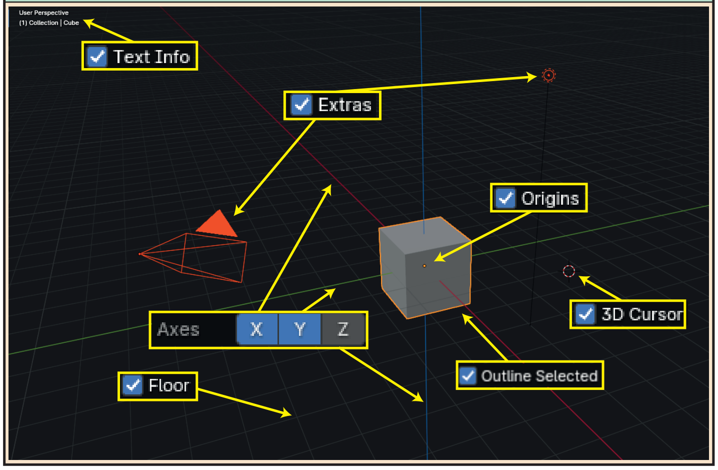
It's the third icon - **Viewport Overlays** - that supplies options which we are more likely to make use of.

Here we can show or hide items such as the grid floor, world axes, 3D cursor, object origins, and the orange outline around the currently selected item.

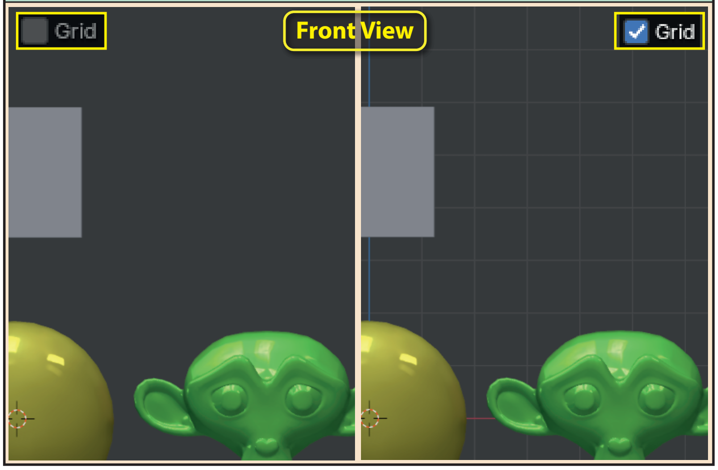
The checkboxes and buttons used to control these and other elements are outlined in yellow here.



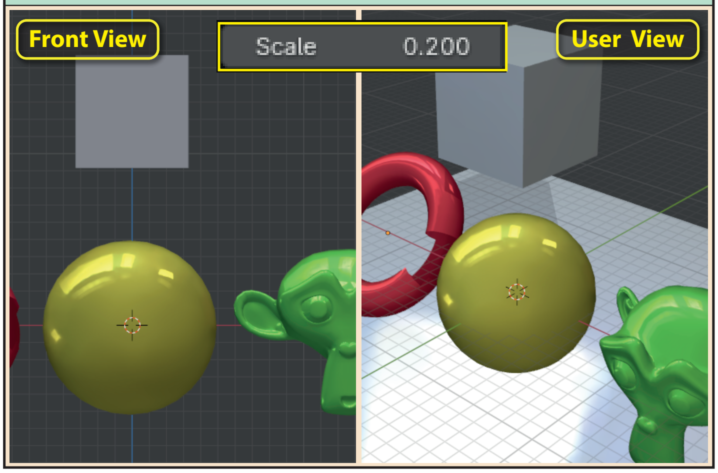
Below we can see the visibility controls matched to their various items.



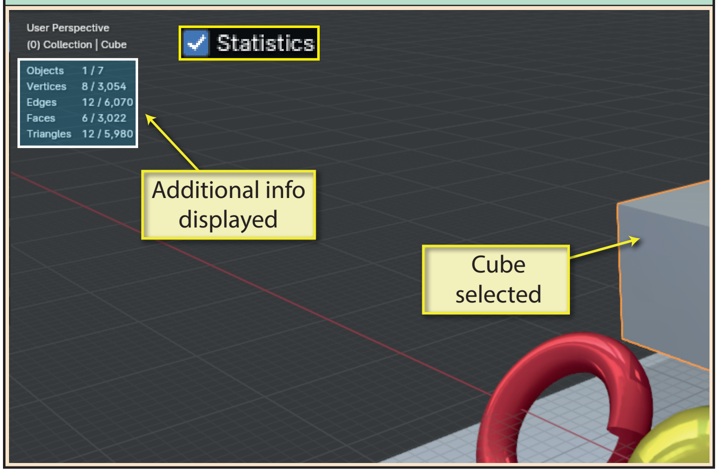
Grid controls the visibility of a grid which is available only when in a named viewpoint such as ***Front***.



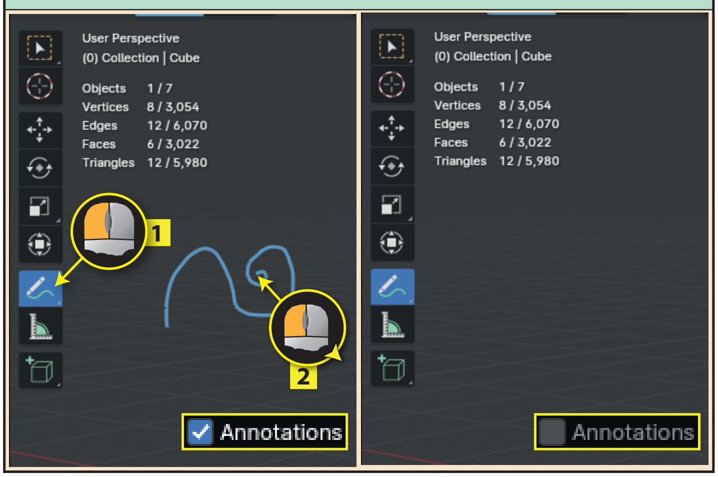
Scale's value determines the size of the squares on the ***Grid*** (if in a named view) or on the ***Floor*** (if in **User** view)



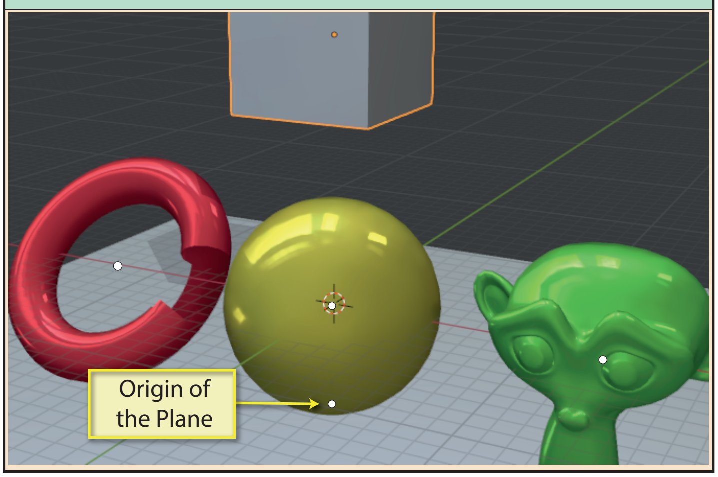
Statistics, if checked, displays additional information about the objects, vertices, edges and faces. The total values for the scene and the number currently selected are shown. The number of triangles are also shown since these are used in video games.



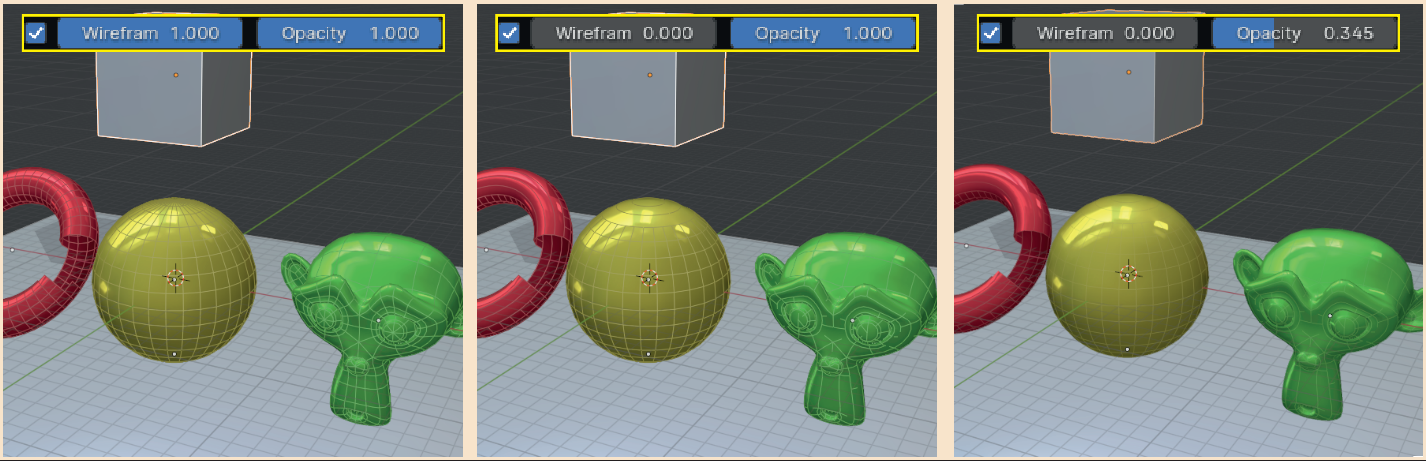
Annotate, when checked, displays any drawings created using the ***Annotate tool*** from the Toolbar.



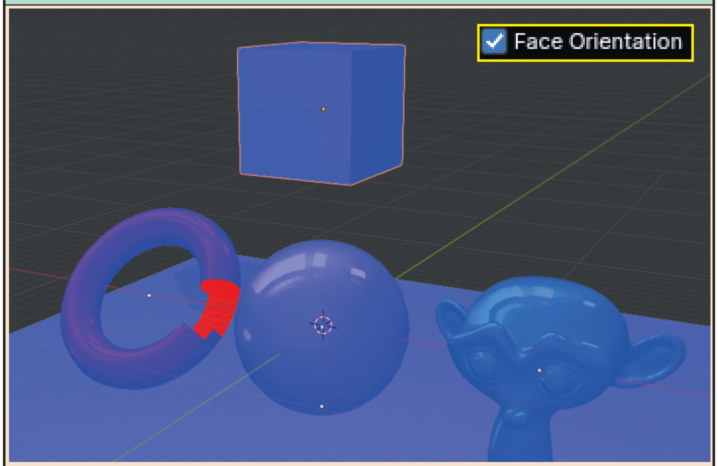
Origins (All), when selected, shows the origin of every item in the scene - even those that are not currently selected. The origins of unselected items are shown in white.



Wirefram[sic] , when checked, displays the edges within a frame. Typically, when the edges are displayed in this way, we refer to the resulting display as ***wireframe***. There are two parameters associated with this option. The **Wirefram** field adjusts how many edges are actually displayed. Lower values remove displayed edges that are common to faces that are at lower angles to each other. **Opacity** adjusts the visibility of all edges. Values close to zero make all edges disappear.

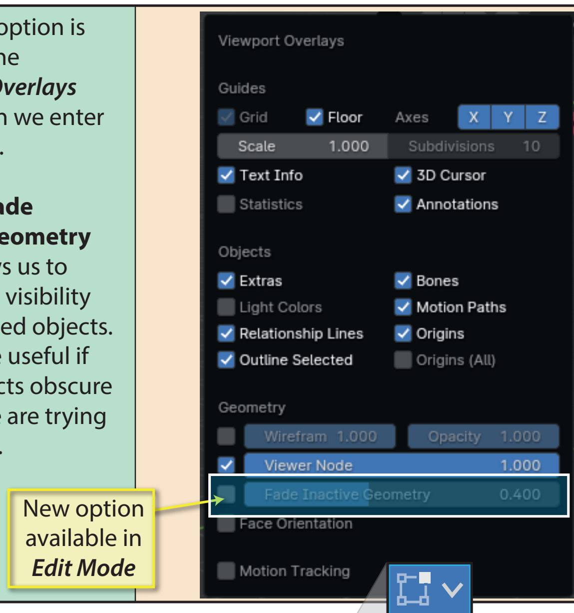


Face Orientation, when checked, displays front faces in blue and back faces in red. Normally, back faces are hidden, but below we can see that some back faces are exposed within the opening in the torus.

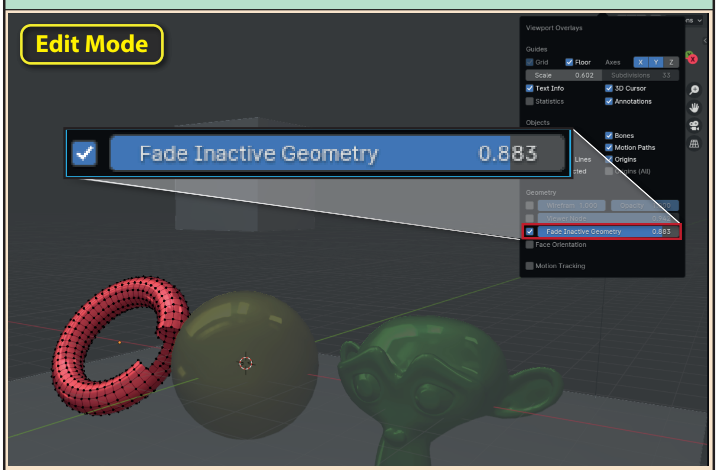


One more option is added to the ***Viewport Overlays*** panel when we enter **Edit Mode**.

The new **Fade Inactive Geometry** entry allows us to control the visibility of unselected objects. This can be useful if these objects obscure the one we are trying to work on.



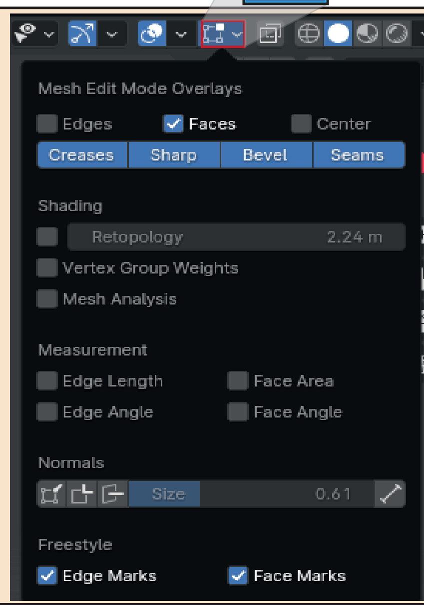
In the example below, the ***Torus*** has been selected before entering ***Edit Mode***. The **Fade** option has then been used to reduce the visibility of other elements.



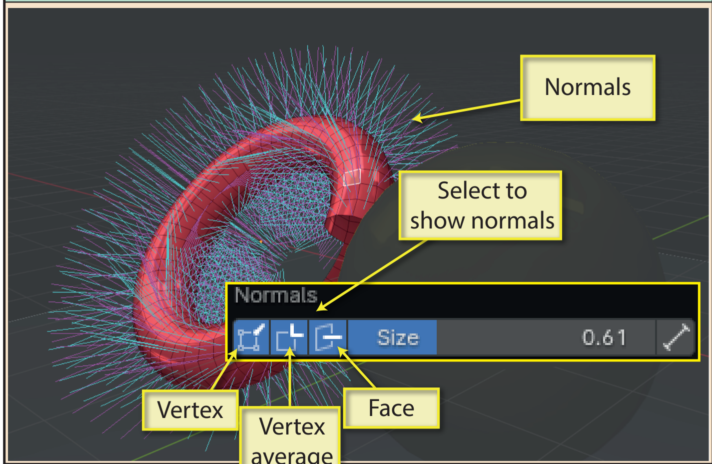
Another result of entering ***Edit Mode*** is that we are presented with an additional entry in our block of icons.

This is the **Edit Mode Viewport Overlays** icon with its advanced set of options.

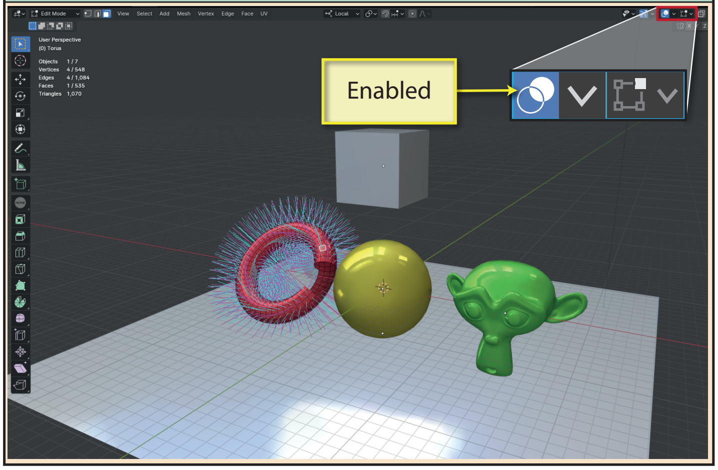
Most of these can be ignored at this time, but one item of interest is ...



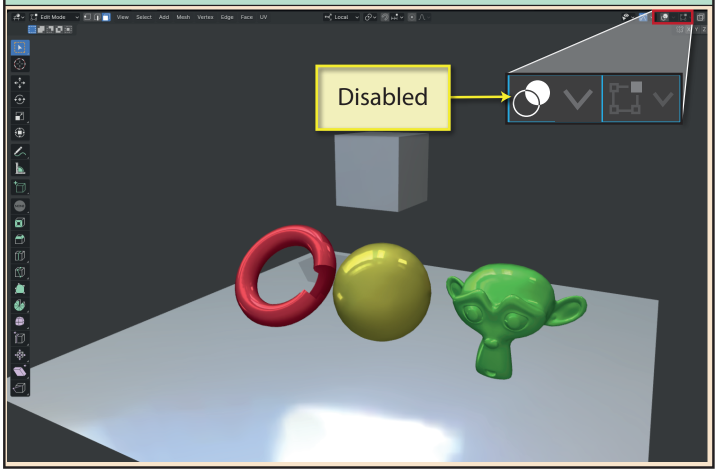
...the **Normals** option. This allows us to make normally invisible **face**, **vertex**, and **vertex average** normals show on the screen. Each normal type is colour-coded. The value to the right adjusts the length of the normals.



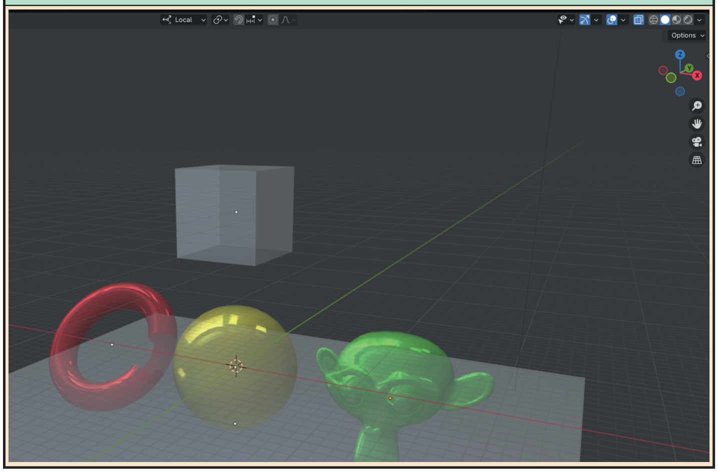
Like the ***Gizmo Visibility icon***, the **Viewport Overlay** icon can be deactivated by clicking on it to disable every option that has been selected within its panel. Below we can see the display when the options are active.



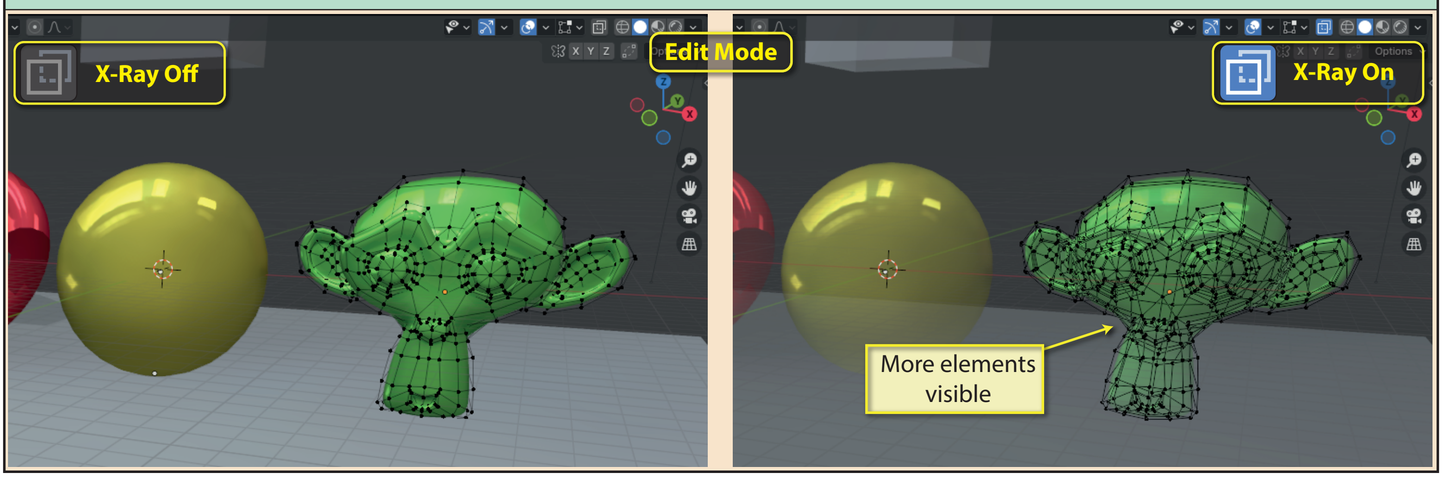
When the ***Overlay Visibility icon*** is disabled, the ***Edit Mode Overlay Visibility icon*** is also automatically disabled.



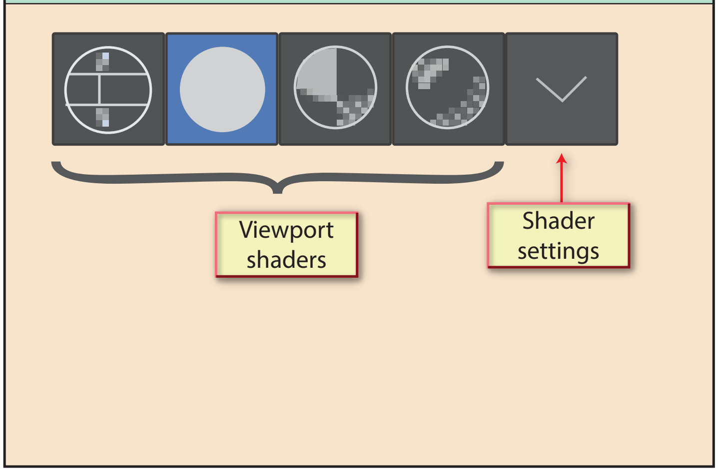
The next icon is the **X-Ray Mode icon**. This icon - which has no dropdown panel - adds a level of transparency to all the objects in our scene as shown below.



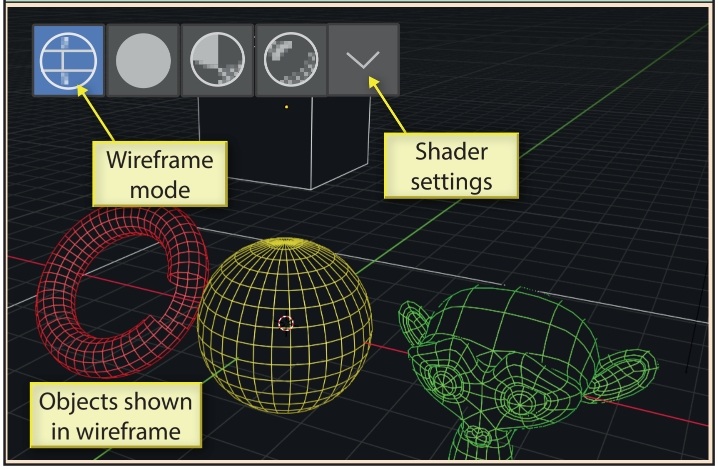
However, ***X-Ray Mode*** is really designed to be used in **Edit Mode** where it allows us access to elements of the currently selected mesh that would normally be hidden from our viewpoint.



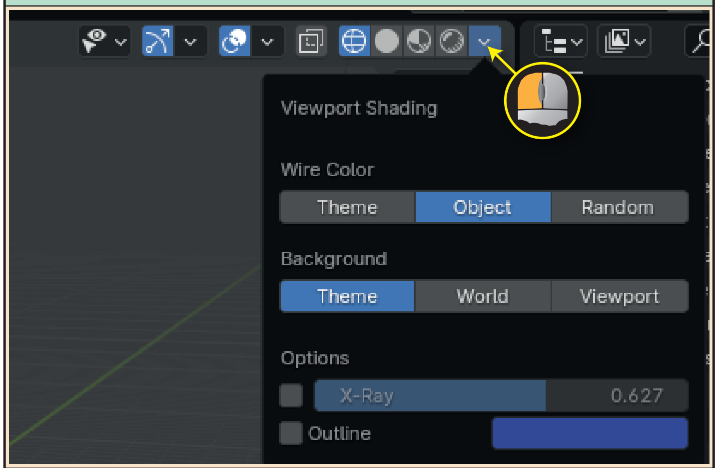
The final four icons control the shading in the ***3D Viewport***. Because they share a common purpose and are mutually exclusive, they appear together in a group with the ***Shader Settings*** button at the end of the group.



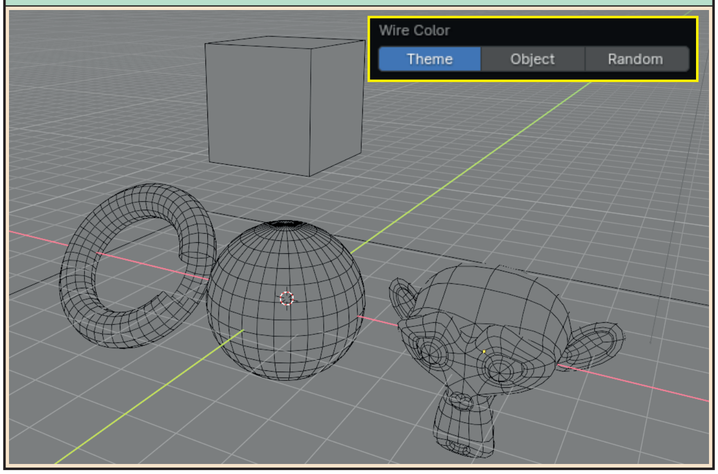
The first of the group displays the scene in **Wireframe mode**, showing the edges that make up the meshes in our scene.



The **Shader Settings** button creates a panel with several entries as shown below.

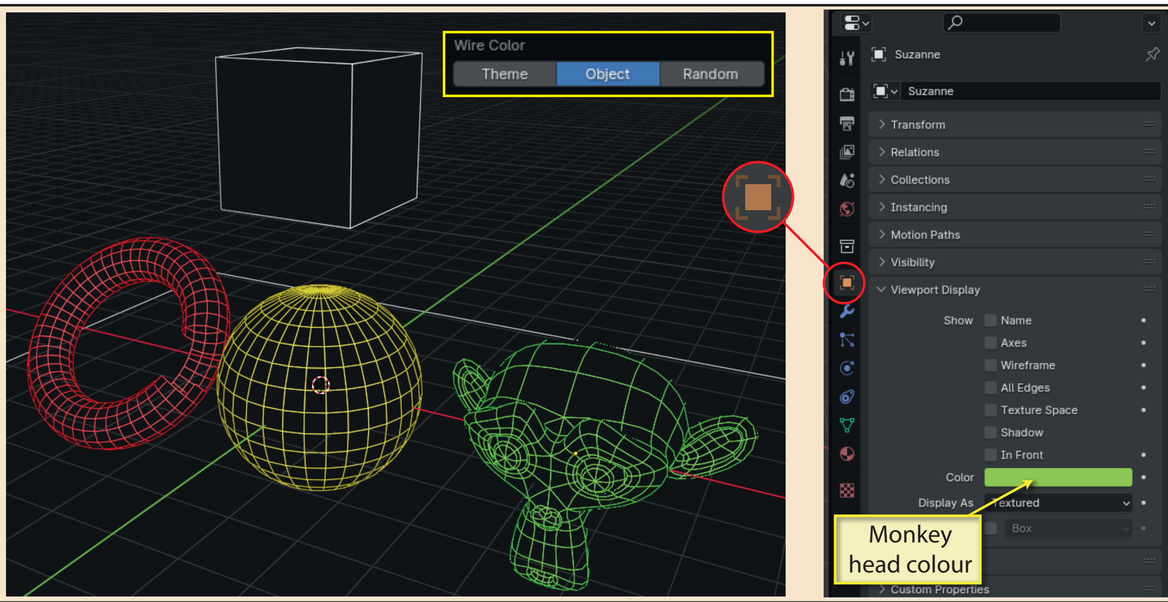


Wire Color determines the colour of the mesh edges. ***Theme*** uses the colour specified for the Blender theme we selected in the first Splash Screen. All meshes are assigned the same colour.

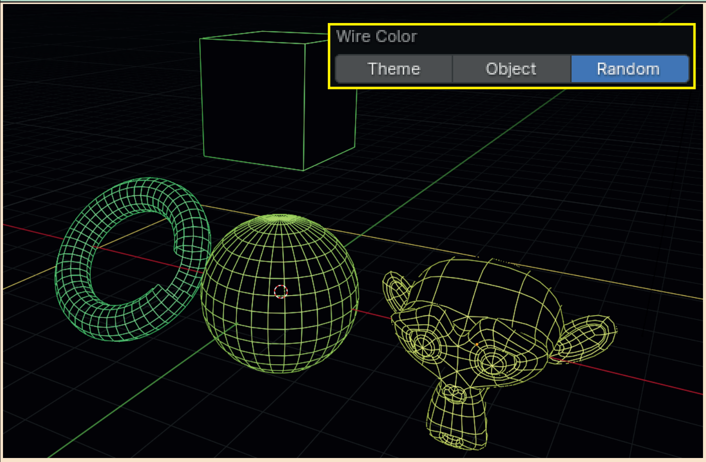


Object colours the edges according to the colour set up for each object in its ***Properties Editor's*** **Object Properties** page.

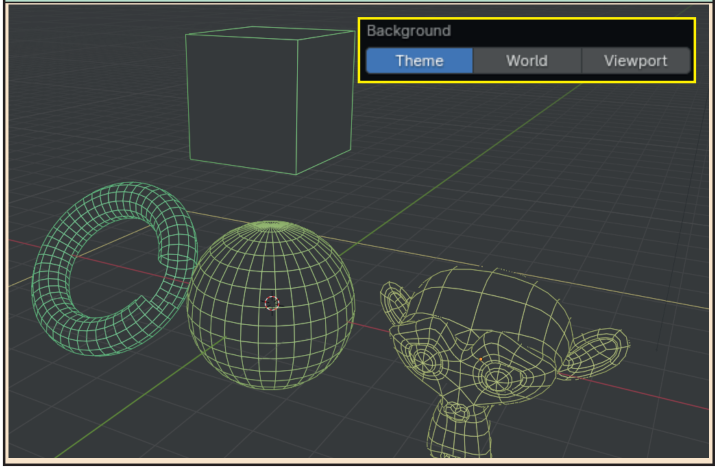
On the right we can see that page for the monkey head.



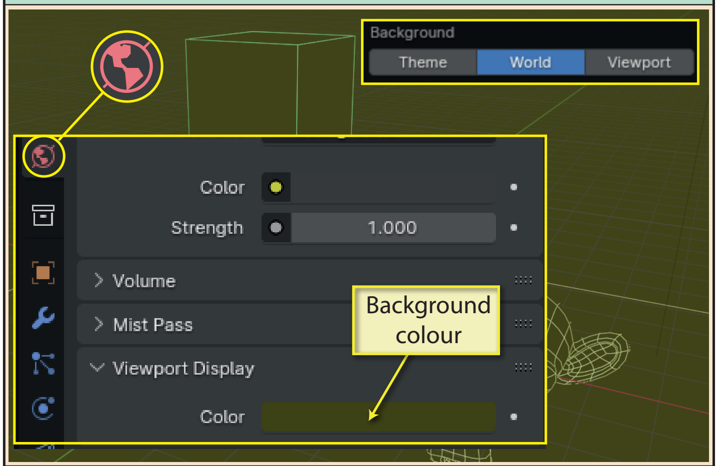
Random assigns a random colour to the each object's edges.



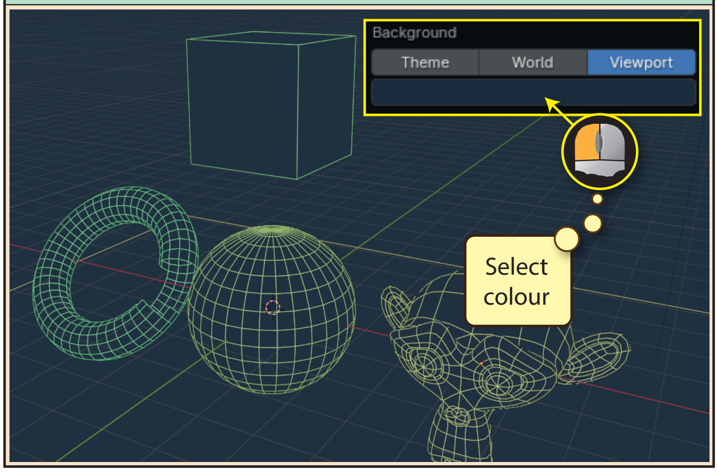
Background is the next heading in the panel. Again, there are three possible values.
Theme uses the colour specified in our selected theme.



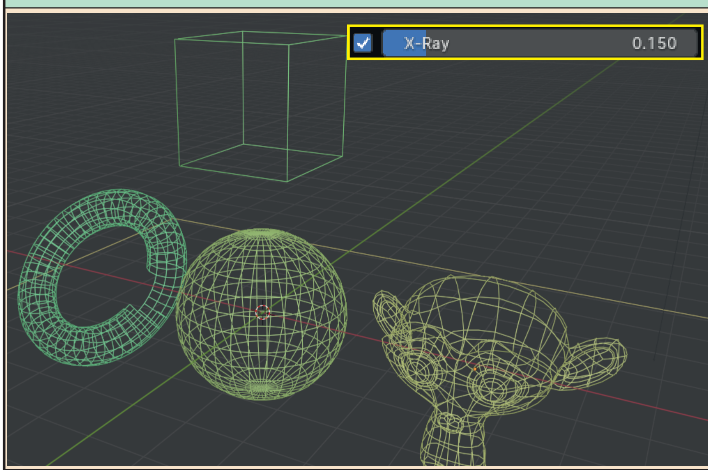
World uses the colour defined in the ***Properties Editor's*** **World Properties page** in the entry ***Viewport Display>Color***.

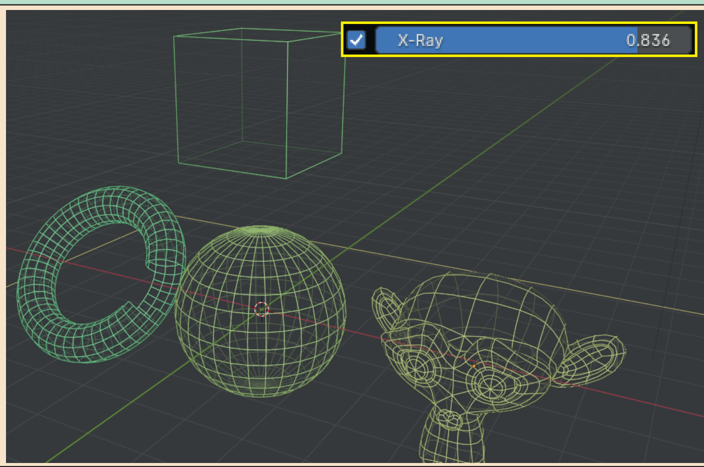


Viewport, when selected, creates a colour bar where we can select any colour for the background.

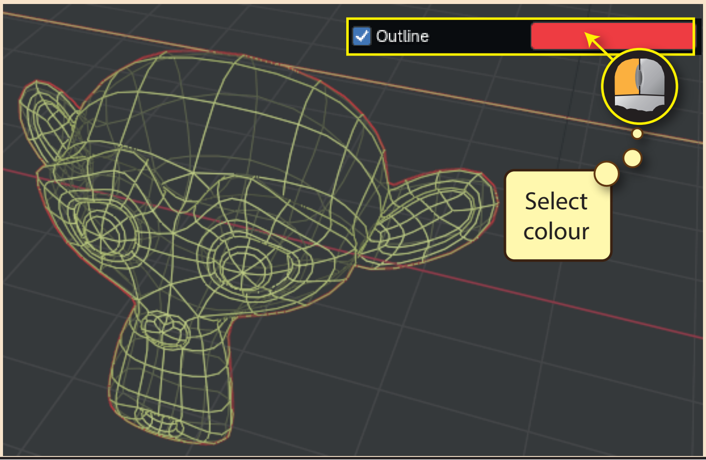


Returning to our ***Wireframe*** parameters panel, the next heading is ***Options***. Here we find **X-Ray**. When switched on, this displays edges that are normally invisible from our viewpoint. The associated value adjusts the opacity of these newly seen edges.

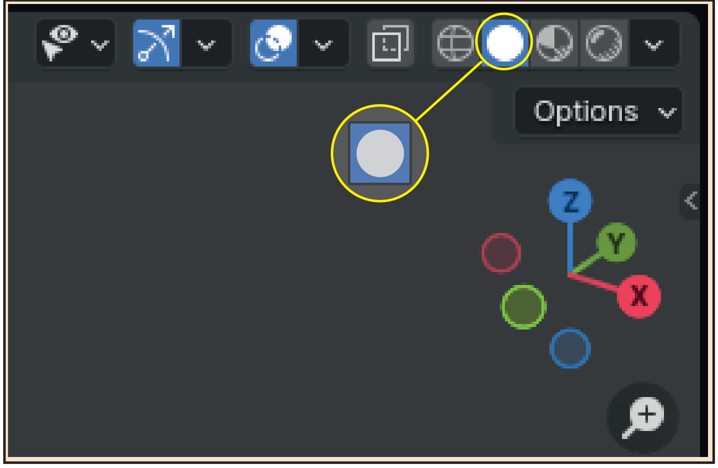




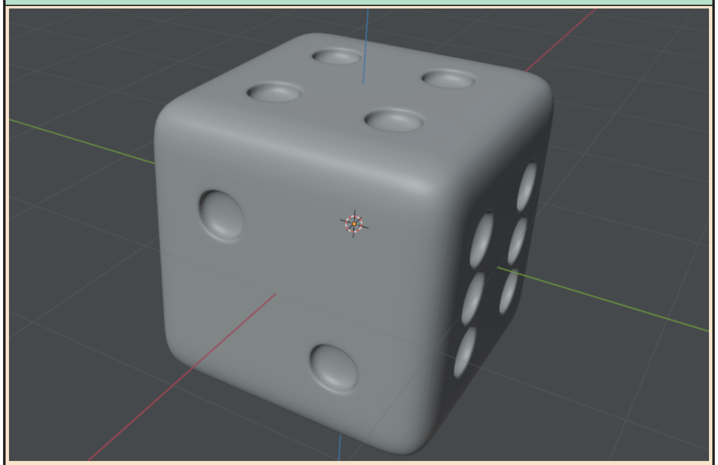
Outline is the final entry in our panel. This sets the outline colour of all meshes. The outline is defined as the set of edges on the border of a mesh as seen from our viewpoint. Here we can see the red outline on the monkey.



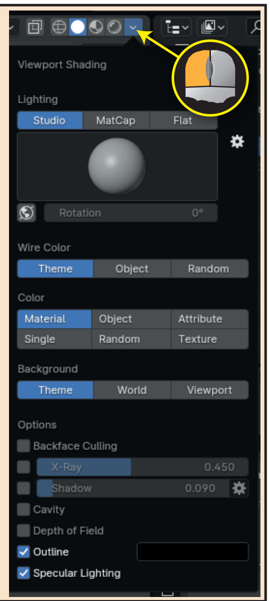
Solid Shading is the next of our ***Viewport*** shaders. This is the default shader used when we start a new project.



If we've not set up any other options in the ***Properties Editor***, the meshes in our scene will appear in shades of grey.

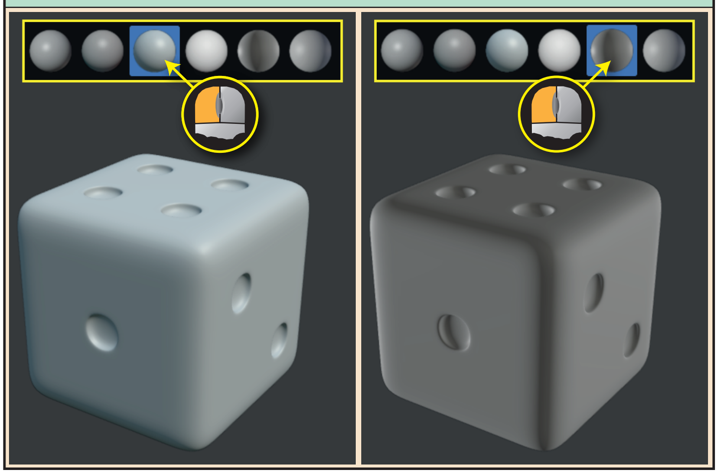


This time, the **Shader Settings** button creates a panel with many more options than we saw in the ***Wireframe's*** panel.

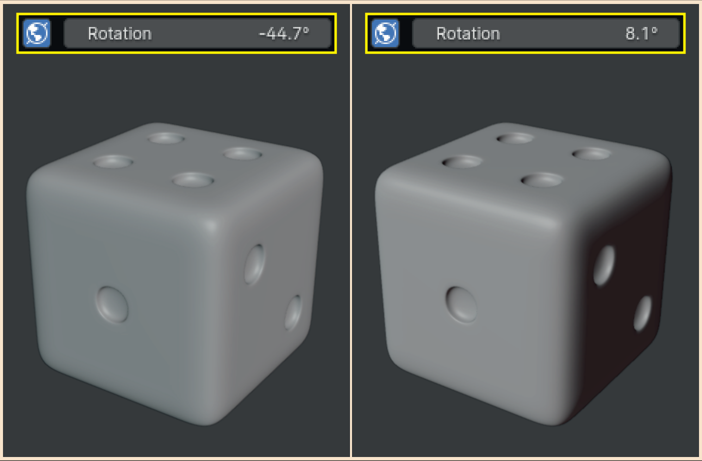


Lighting gives three options. ***Studio*** lighting has, itself, six options, the selected one being represented by the sphere beneath. If we click on this sphere, we'll see all six options in the shape of more spheres.

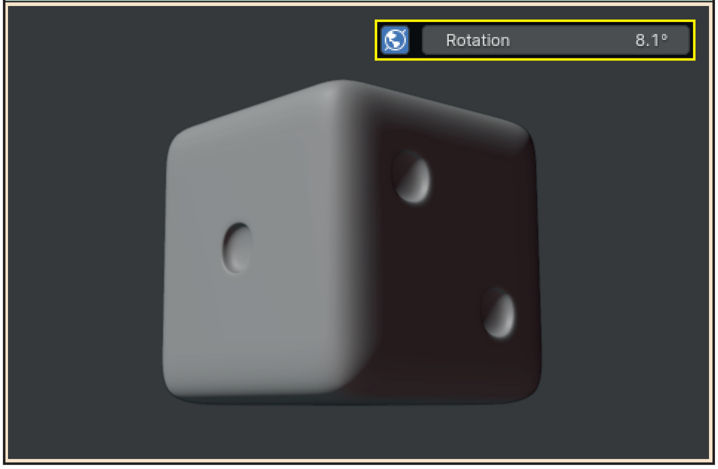
Clicking on one of the spheres will change the lighting on our die. Two examples are shown below.



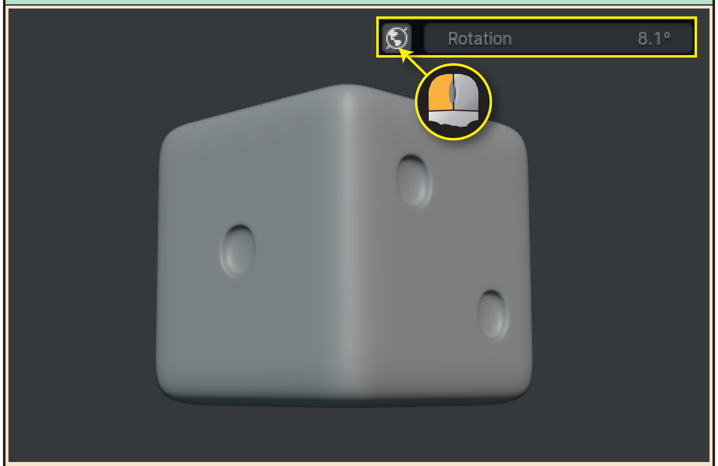
Rotation is a value field beneath the sphere. Adjusting the value here will change the direction of the light and hence the shadows displayed on the objects. Below we can see the effects of different ***Rotation*** values (using second-from-left sphere)



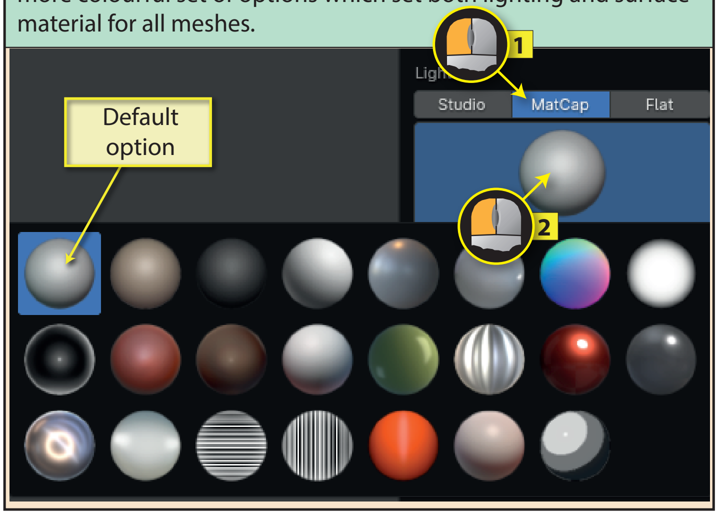
As we move our viewpoint, the light remains consistent and the shadows remain fixed. The example below uses the same 8.1° value as in the last image and the 2 remains in deep shadow as our position is moved.



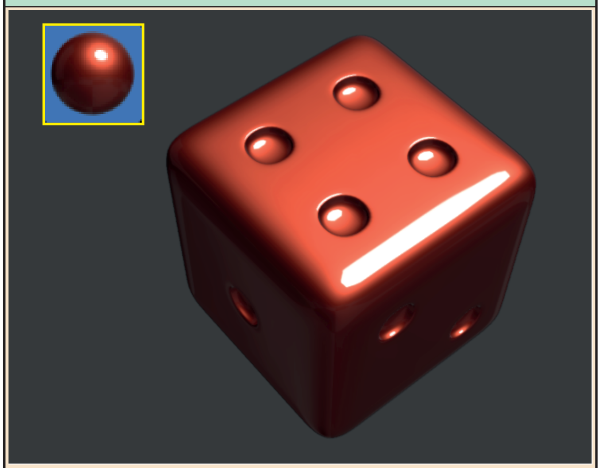
However, if we click on the **Globe** icon to the left of ***Rotation***, to disable it, the shadow moves as our viewpoint changes and the ***Rotation value*** is no longer used.



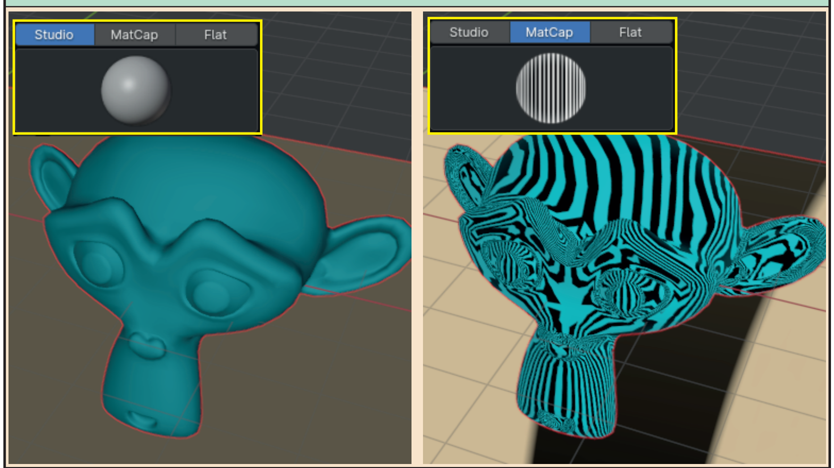
MatCap (short for ***Material Captures***) is the next ***Lighting*** option. Again, the sphere beneath, when clicked displays a larger and more colourful set of options which set both lighting and surface material for all meshes.



If we pick a more dramatic option, we can see how this affects the die.

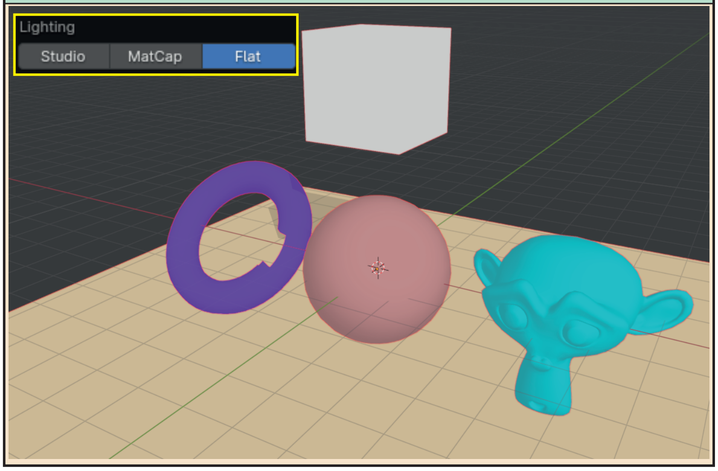


Note that if we've assigned other ***Viewport*** colours to a mesh (we can do that in various pages of the ***Properties Editor***), the ***MatCap*** selection is added to that colour. In the example below, the monkey head has been assigned a blue colour and a striped ***MatCap***.

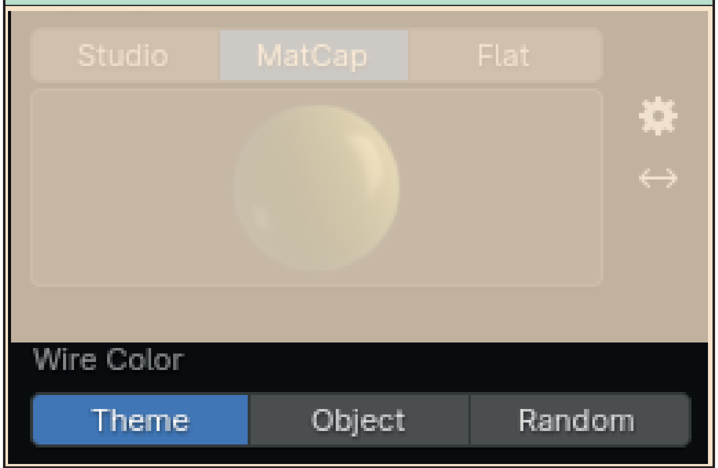


Although we can't rotate a ***MatCap*** light in the same way as we can with ***Studio*** lighting, the doubled arrowed line to the right, when clicked mirrors the effect created as shown on the monkey head below.

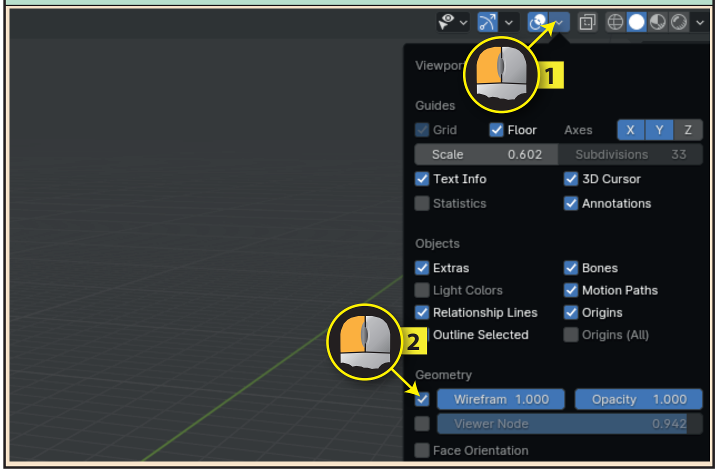
Flat is the final ***Lighting*** option. When selected, no lighting is added and we get a flat, ambient light effect.



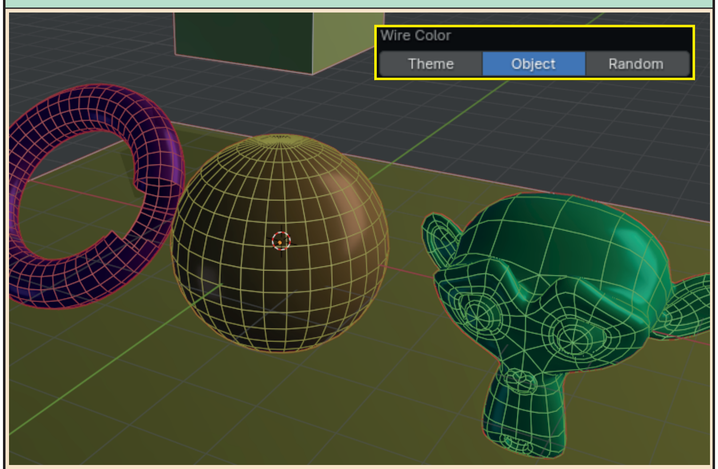
Wire Color, the next heading in the panel, is the same parameter we saw earlier in ***Wireframe Mode***. However...



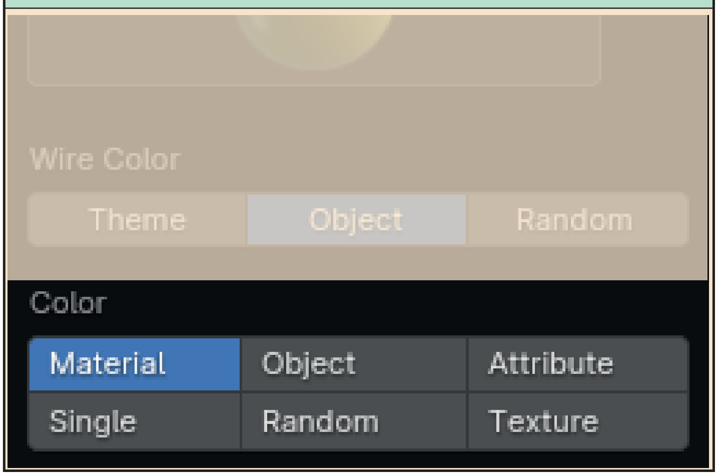
...it is only relevant in ***Solid Mode*** if we have selected ***Geometry>Wirefram*** in the ***Viewport Overlays*** panel...



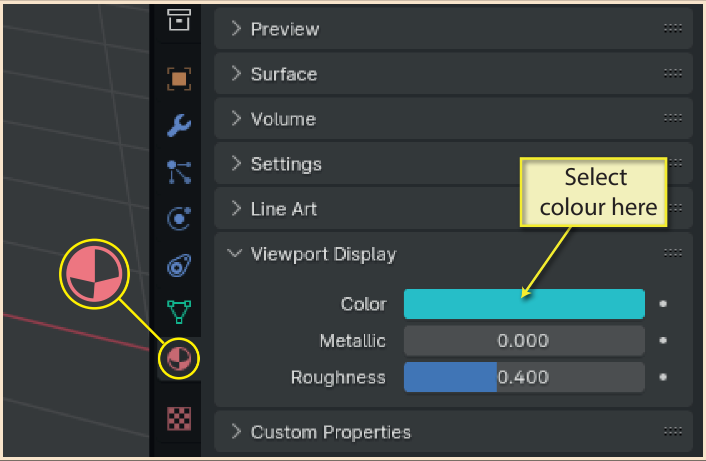
...in which case, the edge will be made visible in the specified colour.



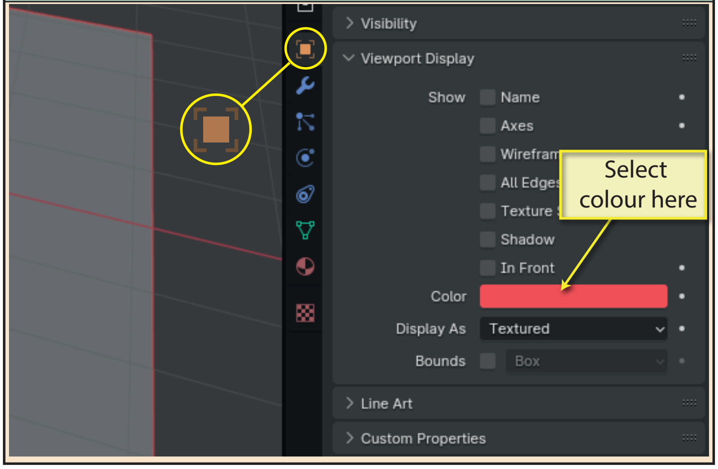
Color, the next heading in the panel, is used to specify which colour the surface of our objects is to display. Although these colours are not designed to appear in the final render they may be useful in other ways during the modelling process.



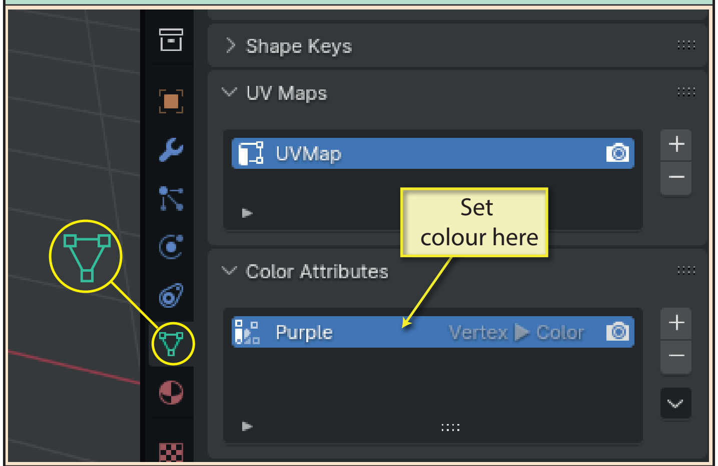
Material, sets the colour of an object to that specified in the ***Materials page*** of the ***Properties Editor***. It can be found under the heading ***Viewport Display>Color***. Below we can see the settings for the monkey head.



Object, when selected causes each object to display the colour defined for it in the ***Object Properties page***, ***Viewport Display>Color***. Below, we see the setting for the torus.



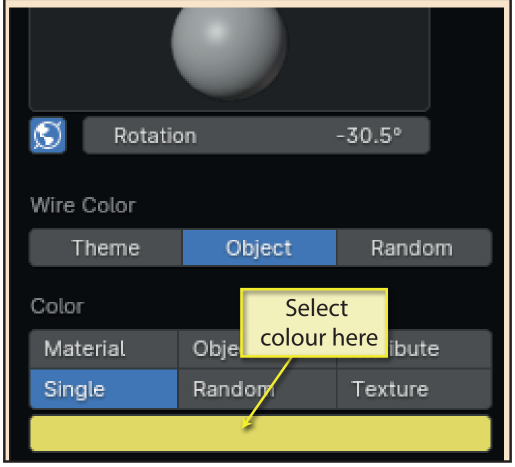
Attribute, sets an object's colour to that defined in the ***Data Properties page, Color Attributes***. How this colour is set up is a little different from the previous two and will be discussed in a later chapter. Below is the setting for the UVSphere.



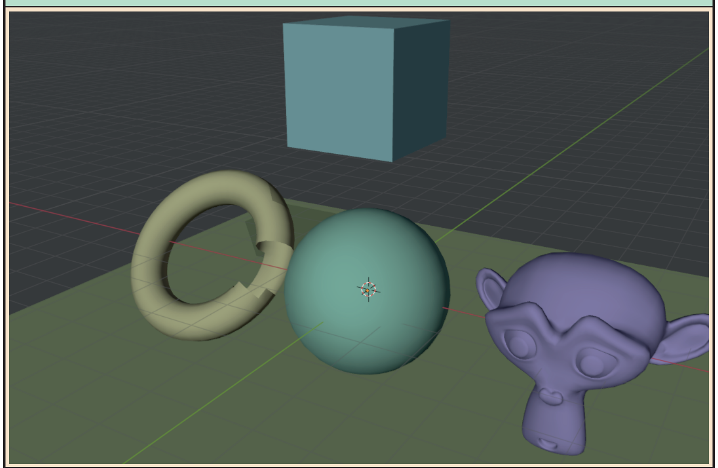
NOTE

If no colour is set in the *Data Properties page*, the colour defined in the *Object Properties page* is shown.

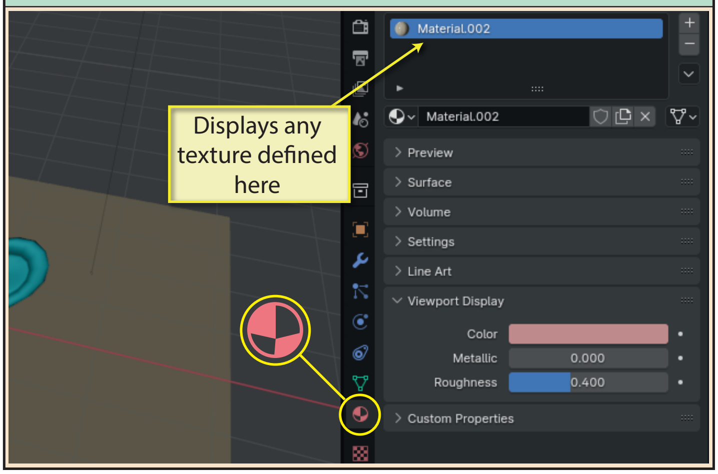
Single shows every object in the same colour. The colour is selected in the colour bar that appears below the six ***Color*** options.



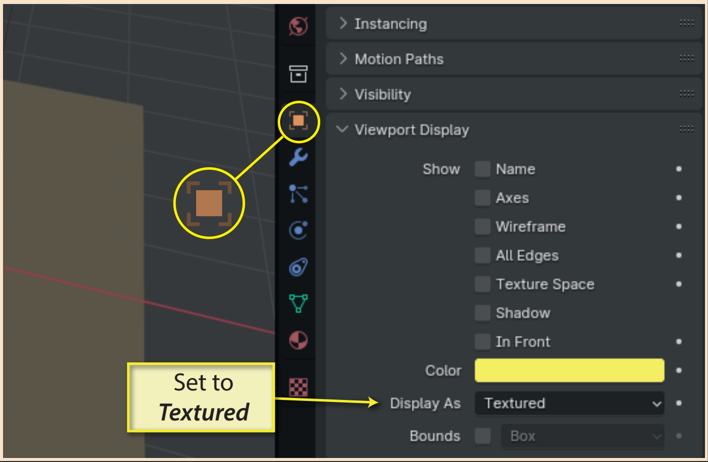
Random shows each object in a different, pale colour.



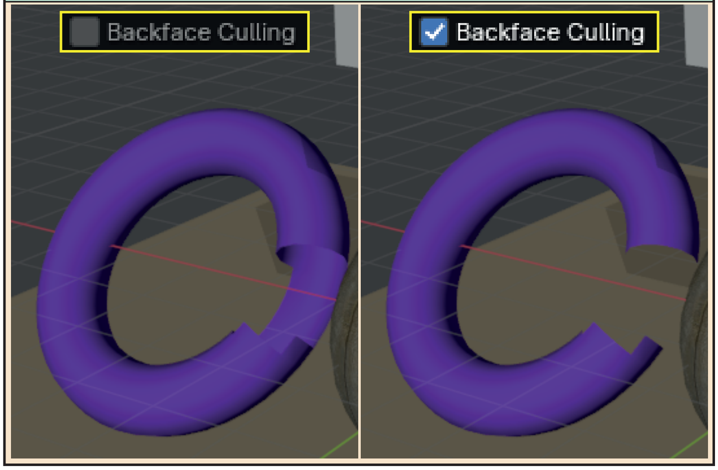
Texture sets the object to display any true texture it has been assigned in the ***Materials page***. If none has been defined it will show the colour defined in the ***Object Properties page***. But...



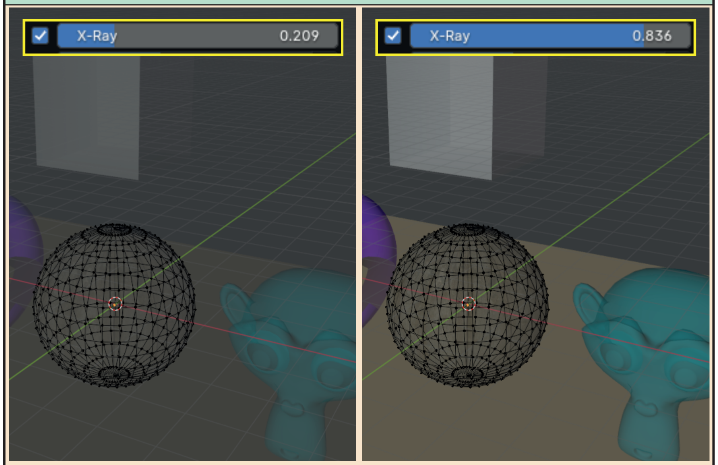
... there is an additional requirement needed before the texture is successfully displayed. The ***Object Properties*** page must have its ***Viewport Display>Display As*** set to ***Textured***.



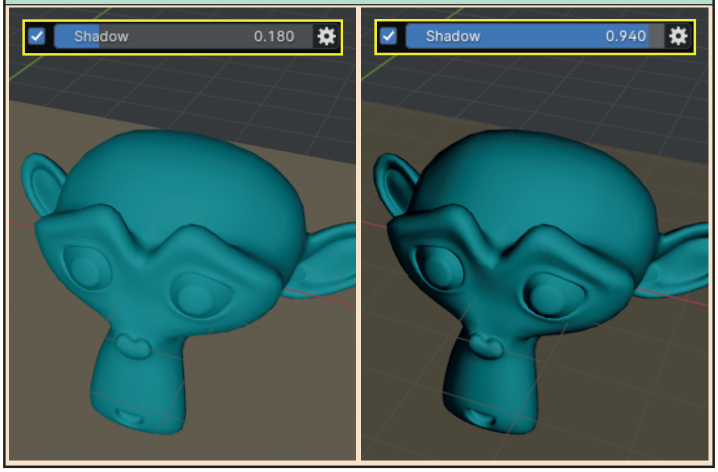
Backface Culling is the next new entry in the panel. When selected this hides all the back faces currently visible. In the scene below, only the internal part of the torus is affected.



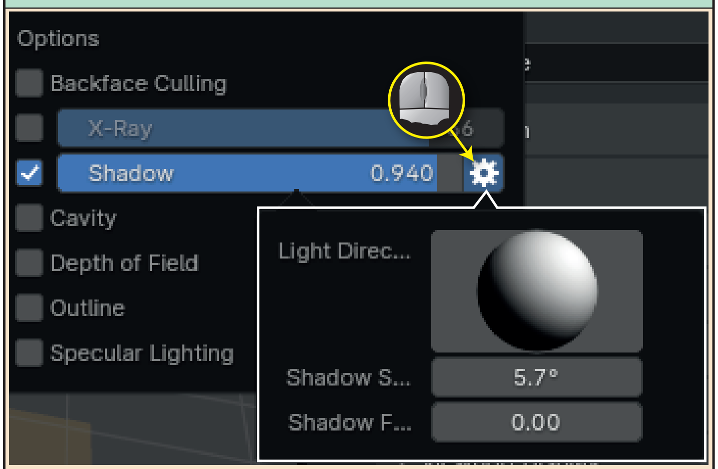
X-Ray, as we've seen before, is best suited for use in ***Edit Mode*** where we can gain access to elements normally hidden from our current viewpoint. The associated value adjusts the visibility of the object's surface.



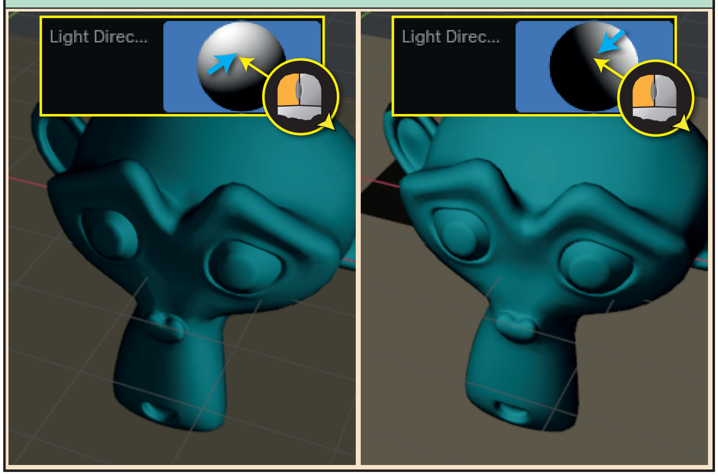
Shadow, when selected, creates a shadow on the surface of each object with the associated value controlling the intensity of the shadow.



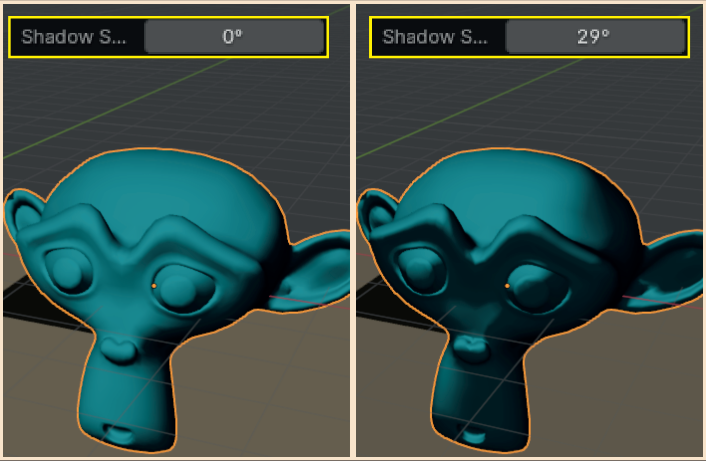
Moving the mouse pointer over the cogwheel creates a small panel offering three more adjustments to the shadow.



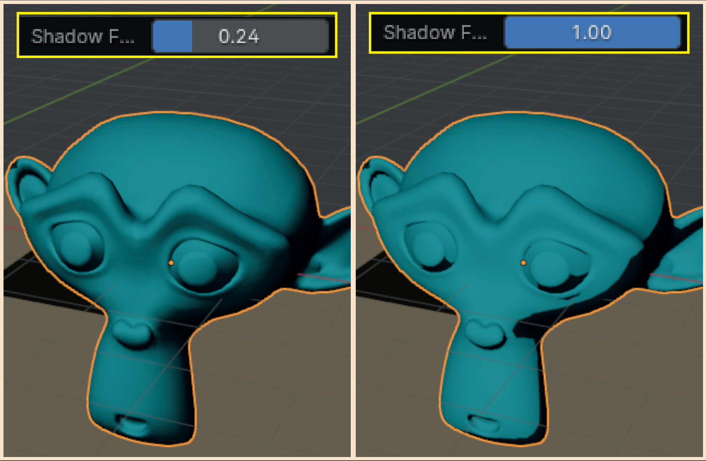
Dragging over the **Shadow Direction** sphere adjusts the direction of the light creating the shadow.



Shadow S[hift] shifts the shadow along the surface of each object.

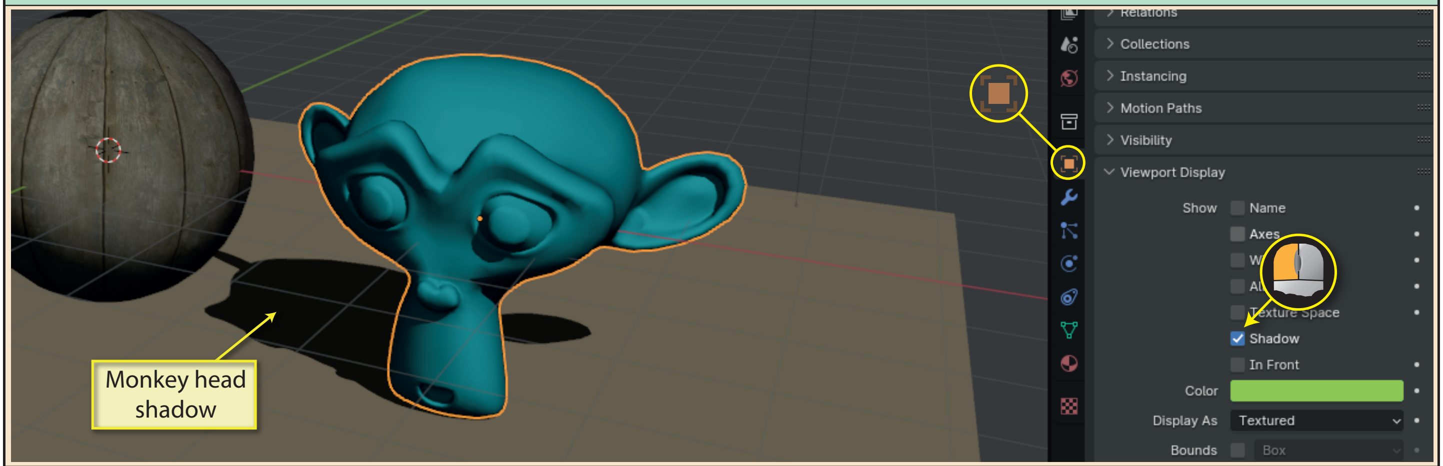


Shadow F[ocus] adjusts the focus of the shadow terminator.

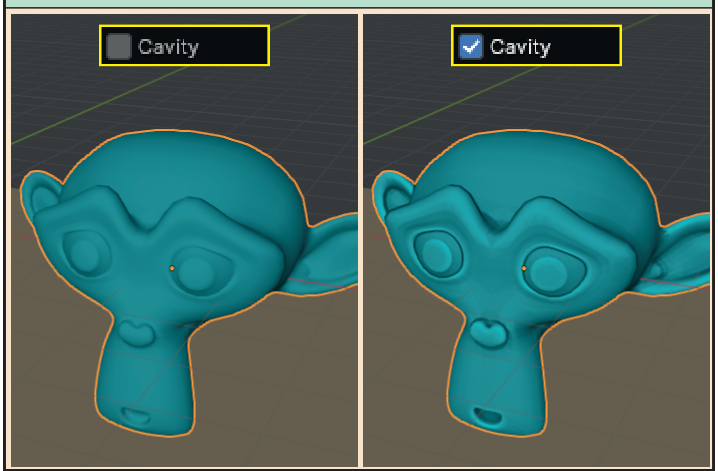


Although we've created shadow on the surface of each object, if we want an object to cast shadows onto other objects within the ***3D Viewport***, then we need to select the object then go to the ***Object Properties*** page and check the box ***Viewport Display>Shadow***.

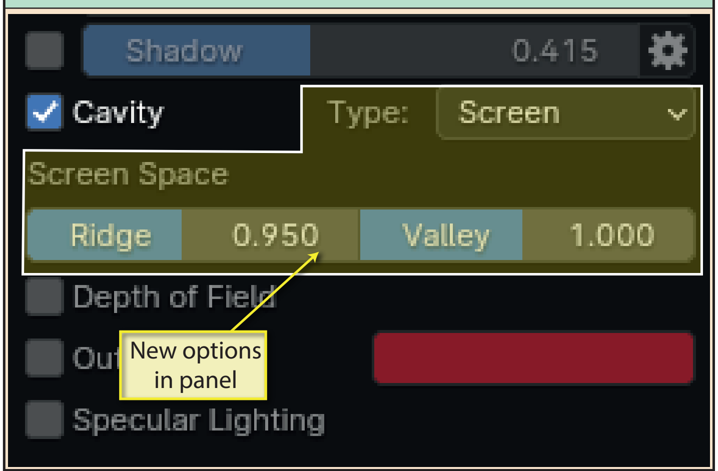
We need to do this for each object that we want to cast a shadow.



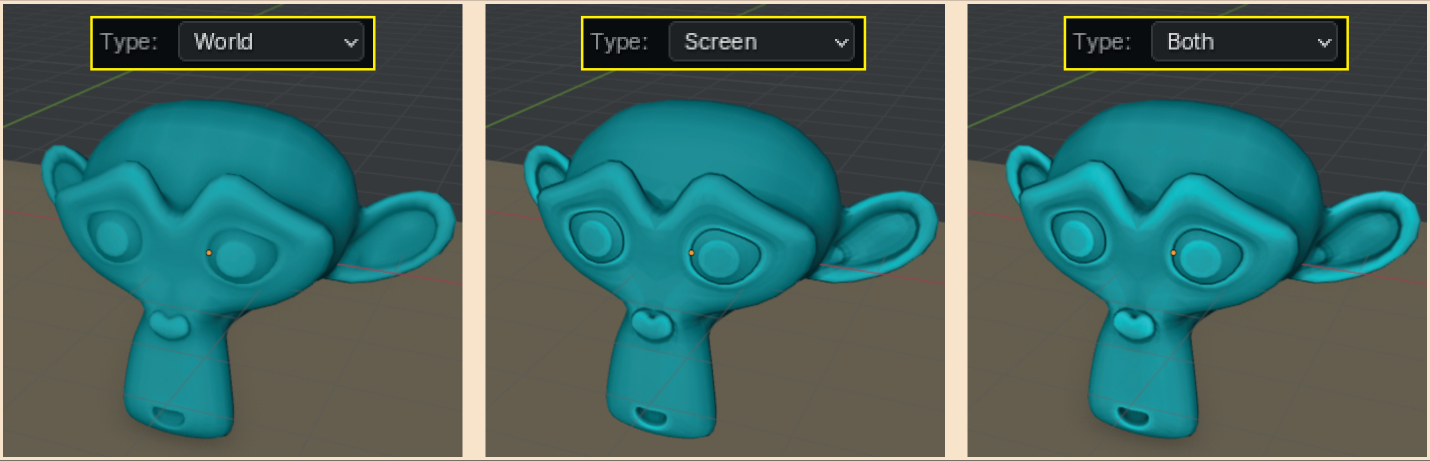
Cavity is the next entry in the ***Solid Shading's*** panel. This option adds shadows and highlights on the surface of an object to emphasise valleys and ridges in all the meshes in the scene.



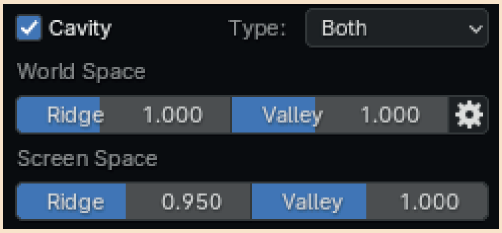
By checking ***Cavity*** we add a few extra parameters to the panel as shown below.



Type offers three options in its dropdown list:
World takes into account the size of the ridges and valleys in the mesh and is the more complex of the two options, taking longer to calculate.
Screen is less accurate but faster. ***Both***, uses both methods at the same time.

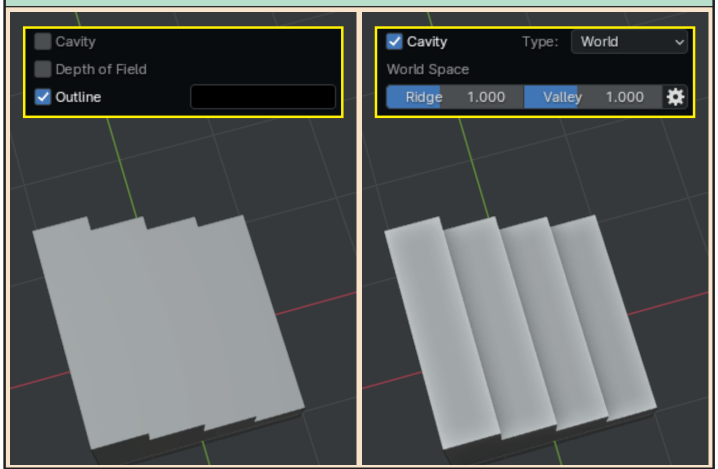


In addition, we have **Ridge** and **Valley** fields to adjust the highlights and shadows created by the ***Cavity*** effect. If we've chosen the ***Both*** option, there are separate adjustments for ***World*** and ***Screen***.

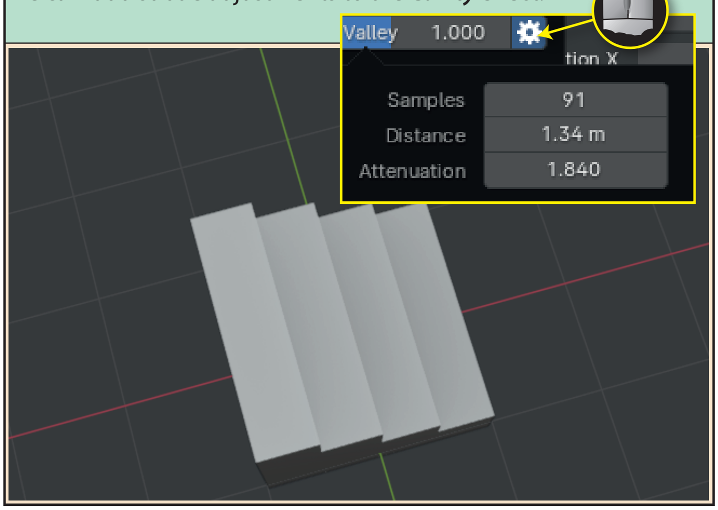


Below we can see the effects created for different ***Ridge*** and ***Valley*** settings when using the ***Screen*** type only.

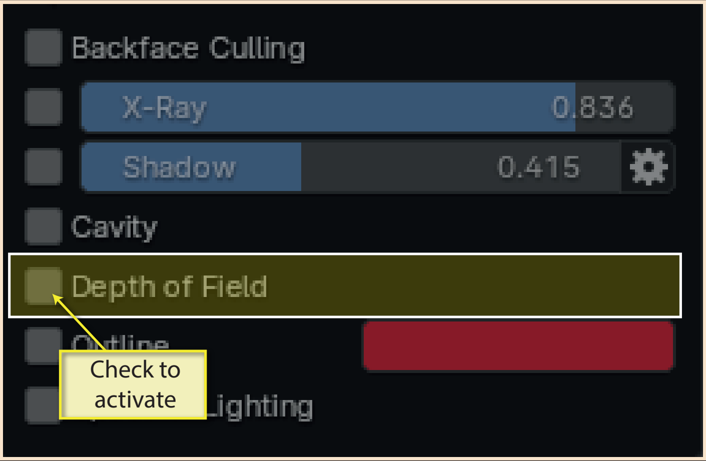
There is a practical use for the ***Cavity*** effect. Below we can see the difference between a set of steps as viewed without and with ***World Cavity***.



When using the **World** option we get an added cogwheel. Moving the mouse over this produces another small panel where we can add subtle adjustments to the ***Cavity*** effect.

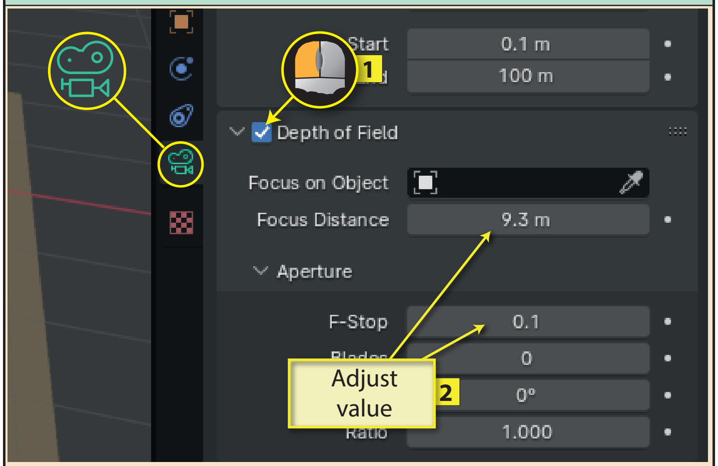


Depth of Field is the next entry in the ***Solid*** panel's settings. This is only relevant when viewing the scene through the render camera (pressing 0 on the Numpad toggles between the render and Viewport cameras).



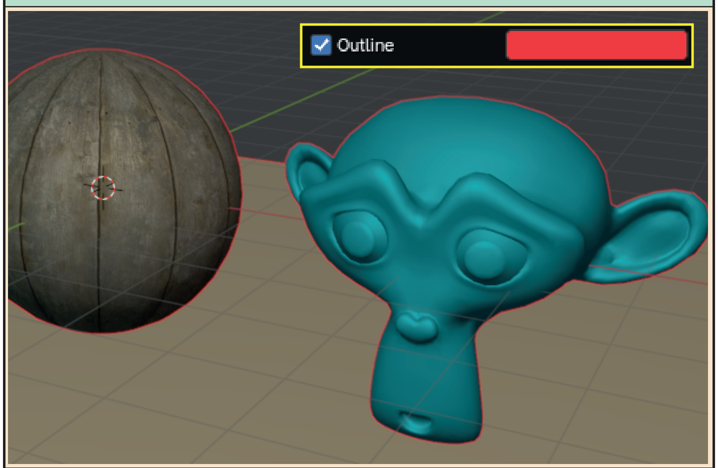
To set up the ***Depth of Field*** we must start by selecting the render camera (click on the ***Camera*** itself in the ***3D Viewport***, or its entry in the ***Outliner Editor***).

Then, in the ***Properties Editor***, we must select the ***Camera's Data Properties page***, check ***Depth of Field*** and reduce the **F-Stop** value to around 0.1 for maximum effect. We may also need to adjust the **Focus Distance**.



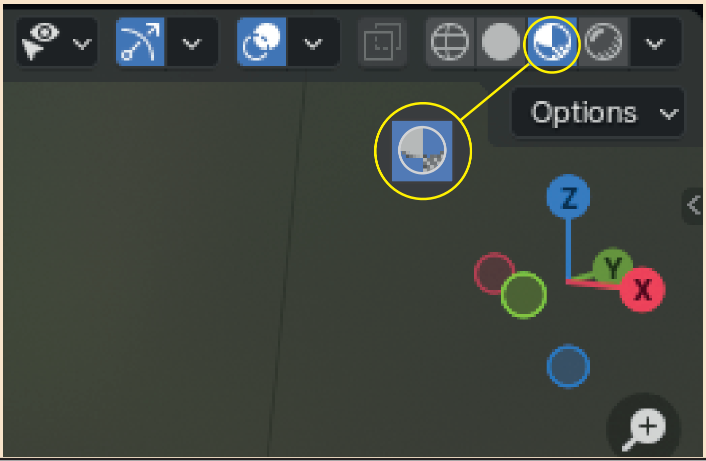
Now, when we change to the render camera view, objects away from the ***Focus Distance*** value will appear blurred.

Outline, we've come across before in ***Wireframe Shading*** and this sets the outline colour for all objects.

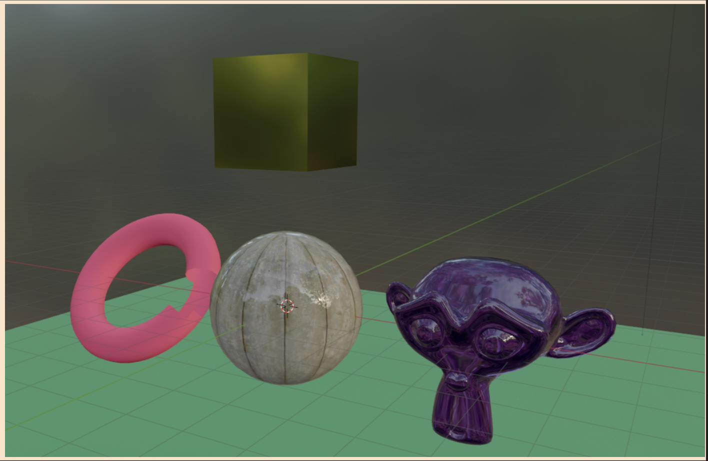


Specular Lighting is only available when ***Lighting*** is set to ***Studio***. When selected, this adds a shininess to the surface of all objects.

Material Preview is the next shading option. As the name suggests, this is the first shader to give a display closer to the final rendered image.



Below we can see how our scene looks after textures have been added to meshes. Blender will use the ***Eevee*** render engine to create the scene.



The adjustments panel for this shading option is shown here.

The main options control the lighting of the scene.

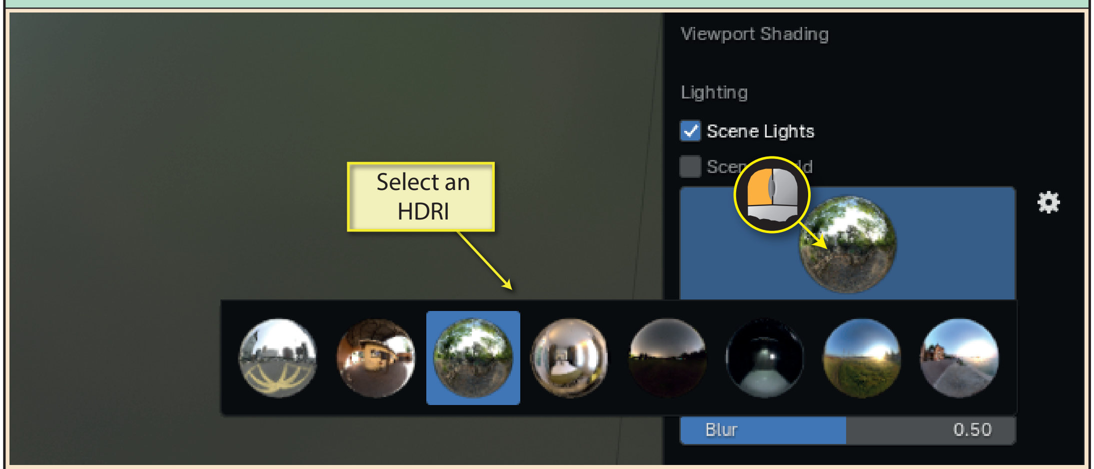
Scene Lights, when checked, include the effects of any light objects that have been placed in the scene. This includes the Point light which is created by default. If this option is unchecked the lights placed in the scene are ignored.

Scene World is the term used to denote the environment surrounding our scene.

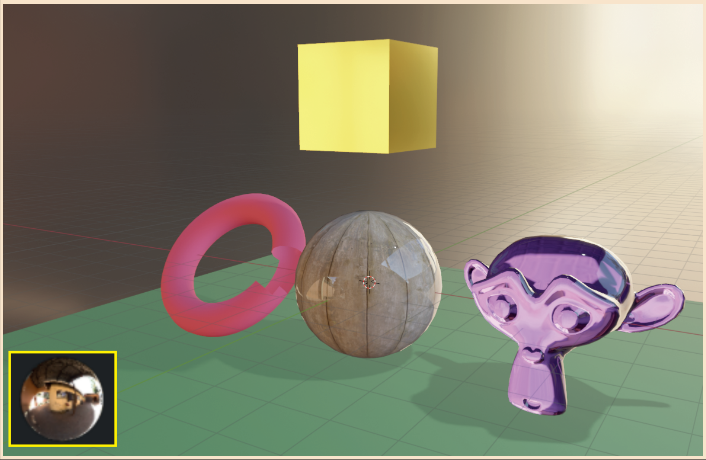
Typically, this is a 360° image referred to as ***High Dynamic Range Image*** or just ***HDRI***.

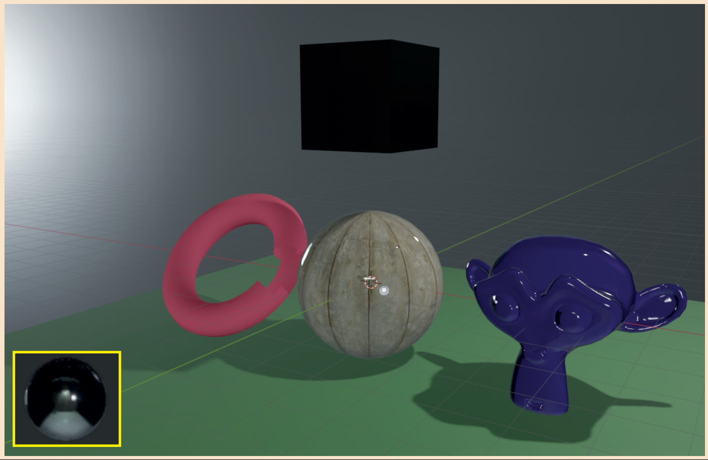
The light that would come from such an environment in the real world is added to our scene to create a more realistic effect.

With the ***Scene World*** box unchecked, the surrounding environment is assumed to be one of the HDRIs available by clicking on the sphere beneath.



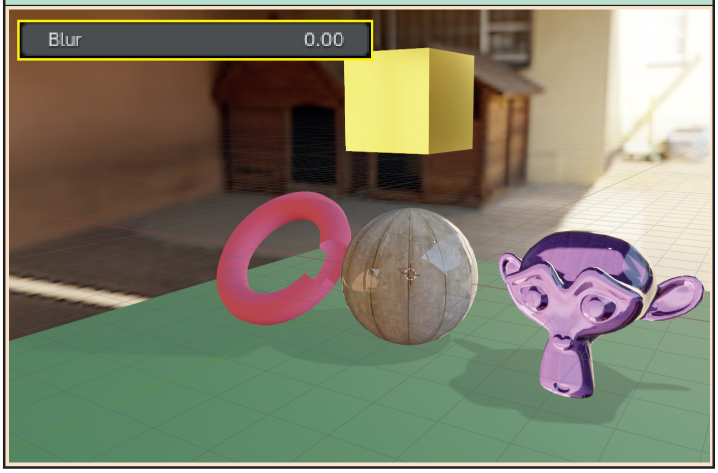
The effect of two of these HDRIs on our scene is shown below.



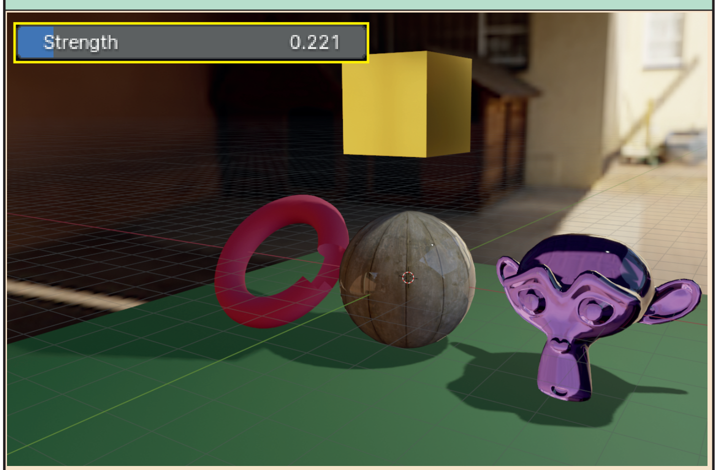


The next group of values controls various aspects of the HDRI. Skipping first to **World Opacity**, this controls the visibility of the HDRI image. If we set the value to its maximum we have a better view of the image used.

Blur adjusts the focus of the HDRI.



Strength adjusts the brightness of the HDRI and hence adjusts the strength of the light falling on our scene.

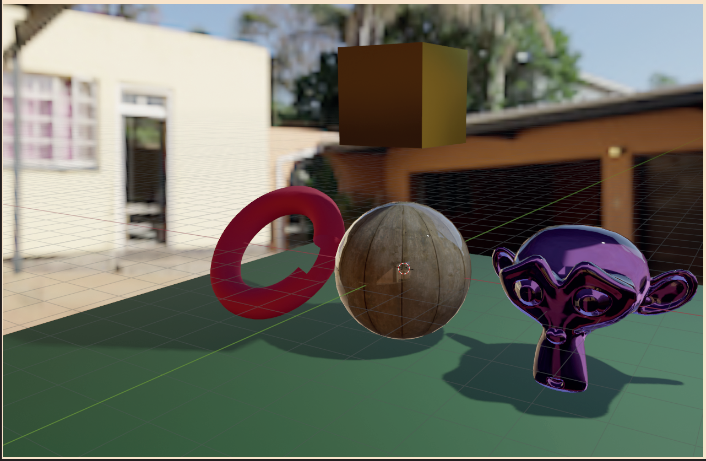


Rotation rotates the HDRI about our scene allowing us to see a different part of the image from our current viewpoint. This also affects the direction of the light coming from the HDRI.

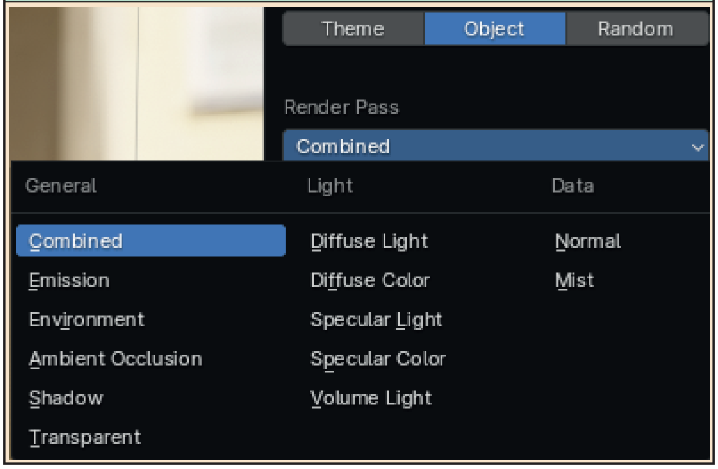
Normally, when we change viewpoint, we'll see a different part of the background image...

...but if we click on the globe to the left of ***Rotation***, the first change is that the HDRI adjusts to its default position, ignoring the rotation setting...

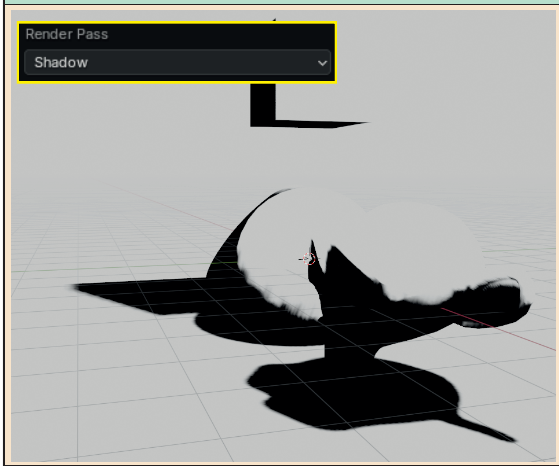
...and as we change viewpoint, the background remains unchanged.

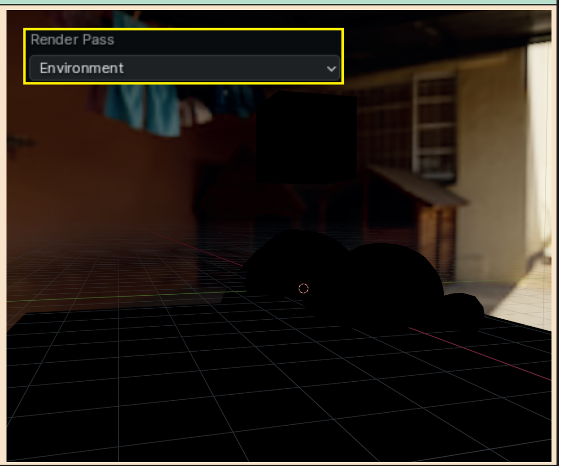


Render Path determines which part of the scene is displayed in the ***Viewport***. This gives us options to display the component parts on their own.



For example, we can use this parameter to show only the shadows created in the scene. Or to show only the HDRI.





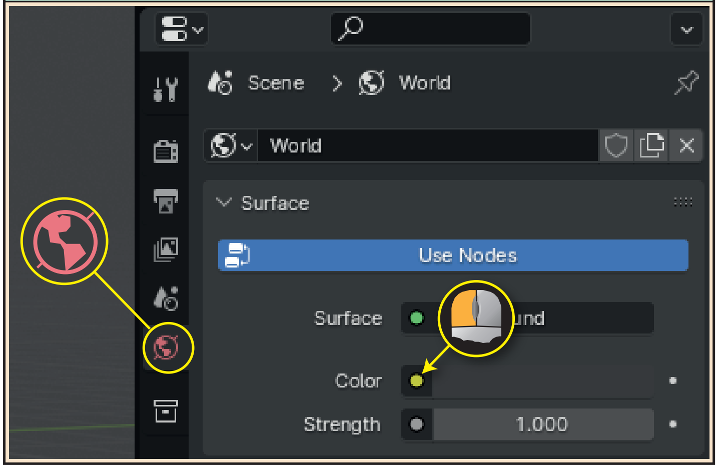
The ***Viewport Shading*** panel's last entry is **Compositor** which controls the availablity of the Blender compositor.

A compositor combines several images into a single image .

This is an advanced topic not covered in this text.

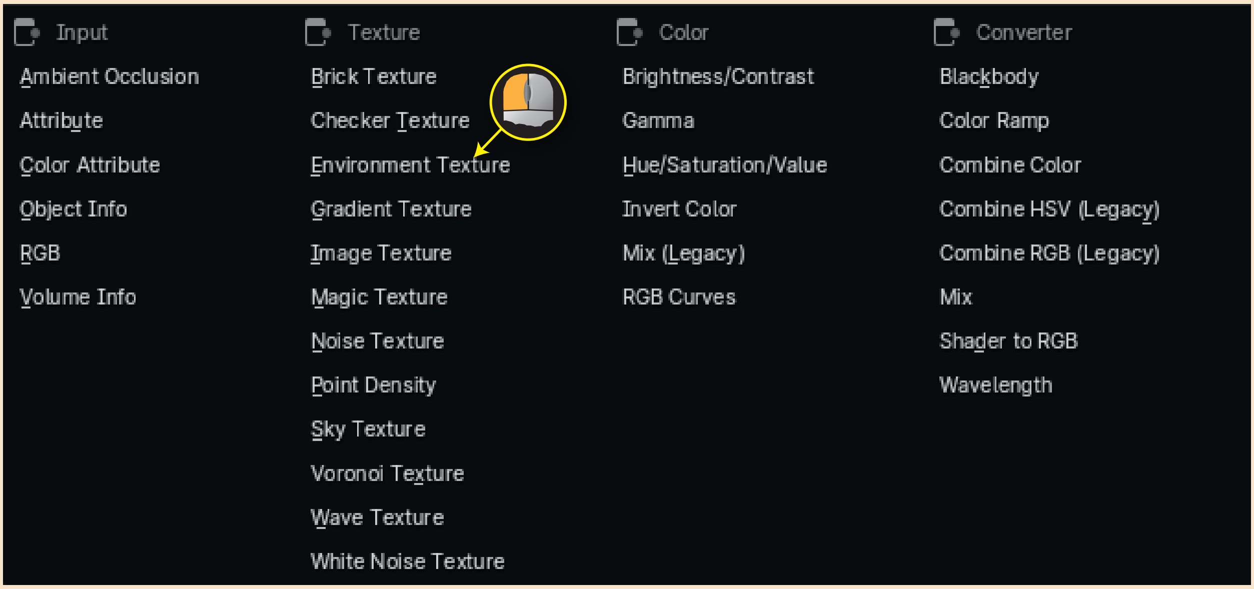
HDRI that we employ through this panel will not appear as background in the final renderd image. If we want the HDRI to be included, we must start by checking the **Scene World** box...

... and then move to the ***Properties Editor's World Properties*** page. There we need to click on the ***Surface>Color***.

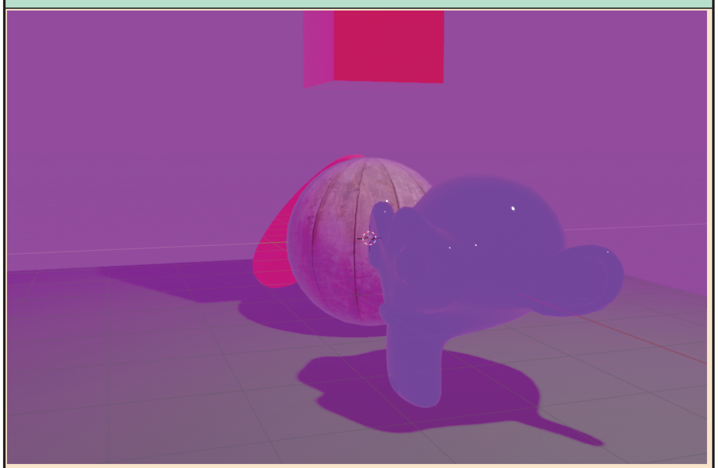


Clicking on that yellow dot produces a panel of options.

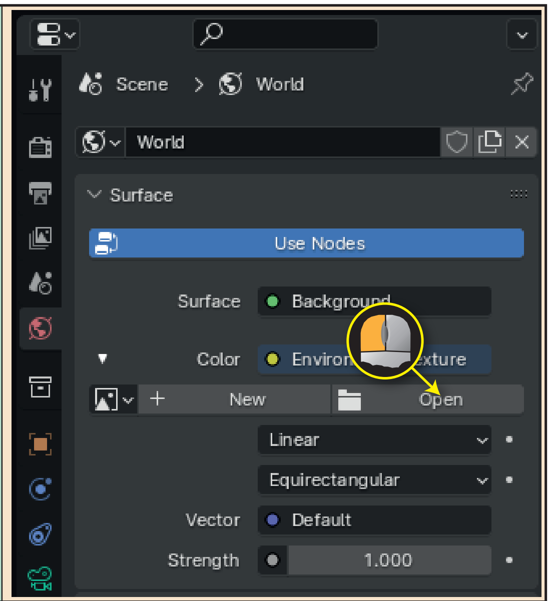
Environment Texture is the option we require in order to add an HDRI.



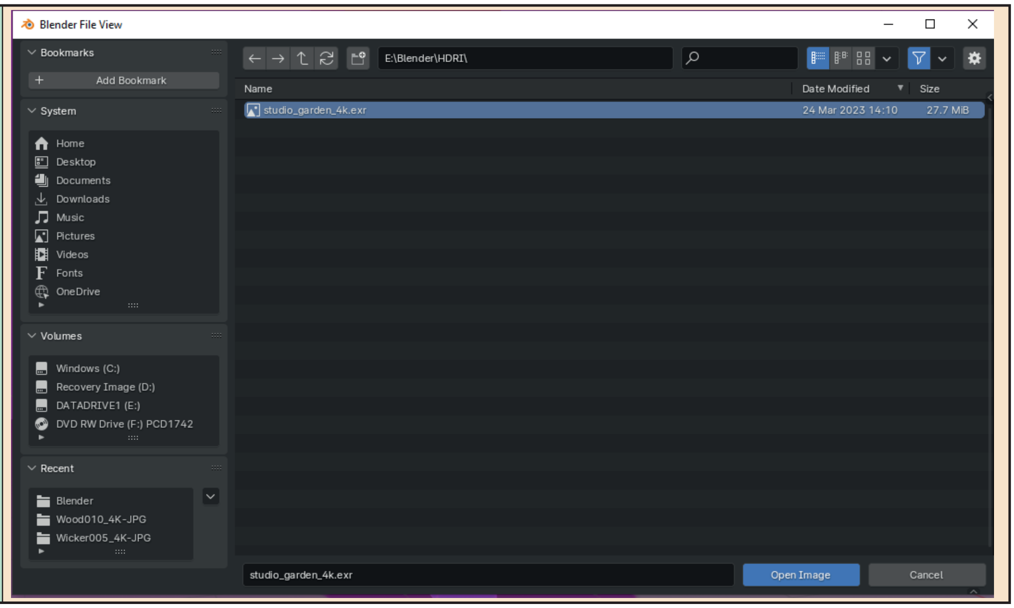
Immediately after selecting this option, the scene will take on a light purple colour. This is Blender telling us that it hasn't yet been given the name of a valid image file.



Now we need to select ***Open*** from the new options on the ***World Properties*** page.



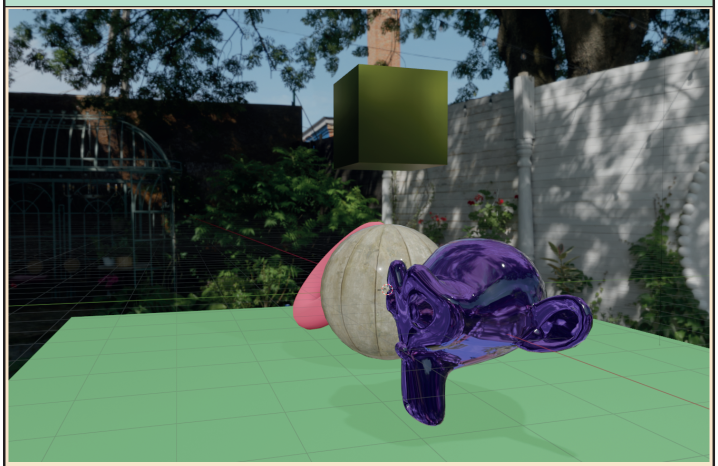
This will produce Blender's ***Open File*** dialog box where we can select the image we want to use as the background.



Note that an HDRI is not a standard image.

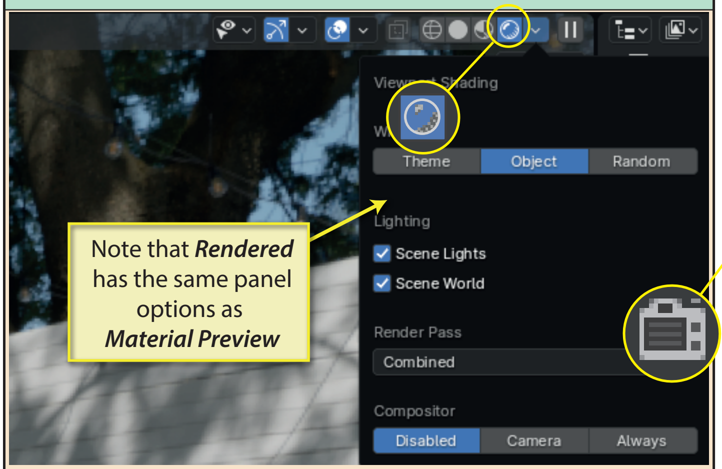
These image files have the extension **.exr** and can be downloaded for free from various websites such as ***polyhaven.com***.

Once loaded, our scene will display the new image in the background.

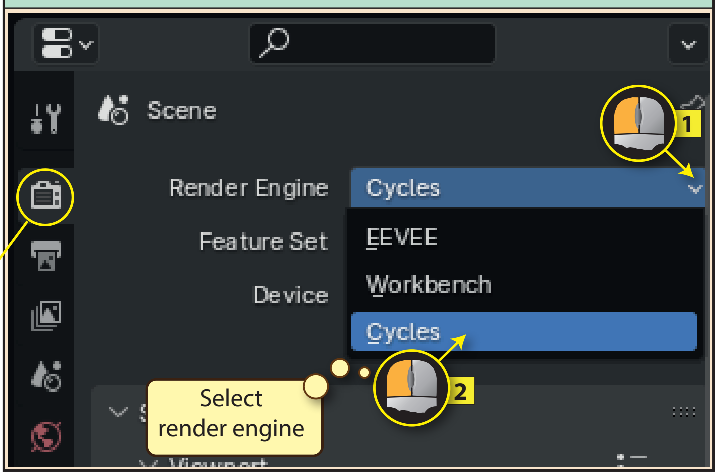


It will also appear in the final render.

Rendered Shading is the final option for the Viewport display. This gives us a result which is close to the final render but from the prespective of the Viewport camera.



In the **Render Properties** page of the ***Properties Editor*** we can select between **Eevee** and **Cycles**. But although ***Cycles*** gives a more accurte result it takes much longer to calculate and will be impractical on all but the fastest machine.



MESHES IN OBJECT MODE

In this section we'll cover the following topics:

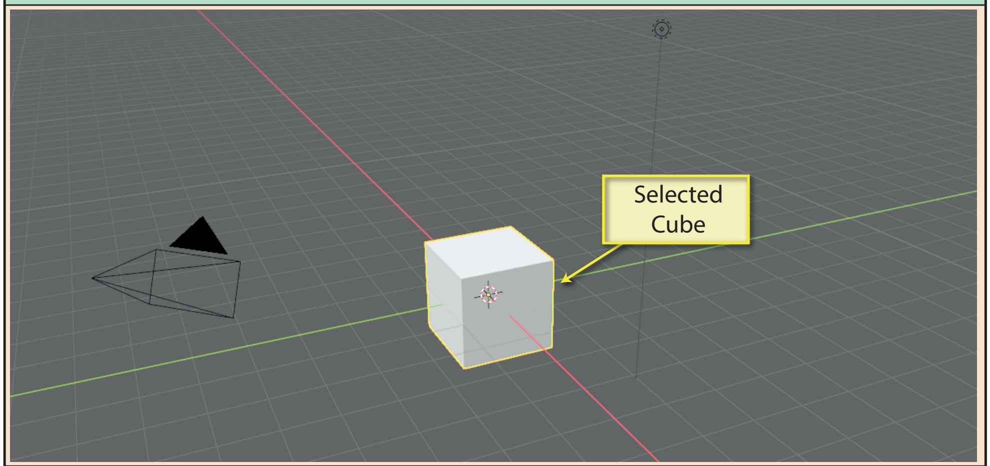
How to create and delete mesh primitives such as cubes, spheres and cones.
How to set the initial properties of primitives.
How to adjust the units of measurement.
The purpose of the 3D cursor.
How to adjust an object's origin.
How to resize, rotate and move mesh objects.

Creating Primitives

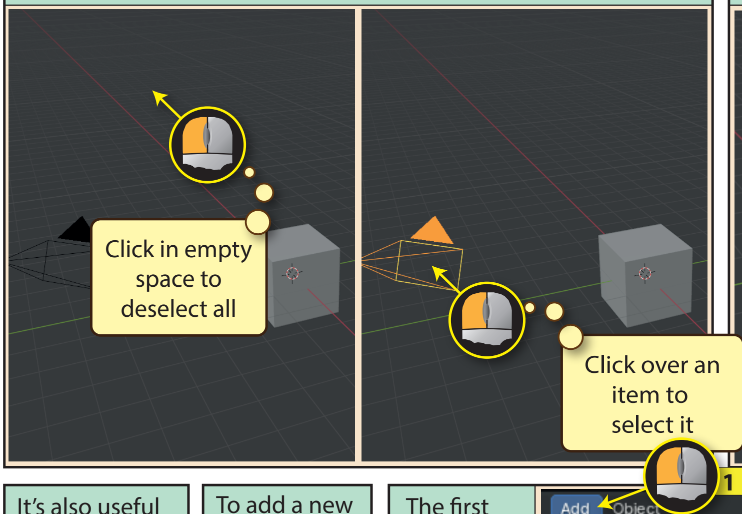
When constructing a scene we often start with one or more primitives.

We have some control over the initial appearance of these primitives so we can adjust some of their characteristics to suit our requirements.

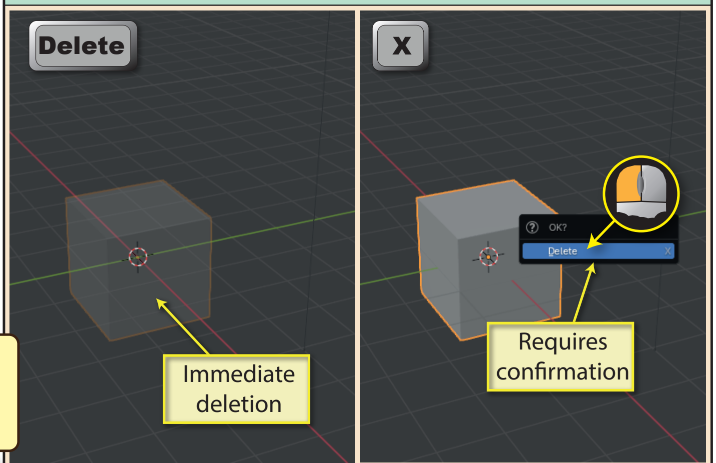
When we start a new project, Blender automatically adds a Cube. We can see from the orange outline that the Cube is also selected.



When working in the ***3D Viewport***, we can deselect an object by clicking in an empty area of the scene.
We can select an item by left-clicking on it.



To delete a selected object, press either the **Delete** key or the **X** key. Using **Delete** deletes the item immediately, but **X** requires confirmation before the deletion is actioned.

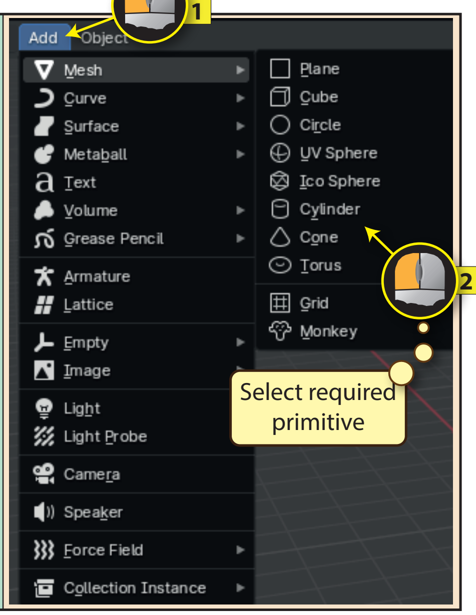


It's also useful to know that any action we perform can be undone by pressing **Ctrl Z**. For example, if we have deleted the Cube but then decide we want to keep it instead, we need only press **Ctrl Z** to have it restored.

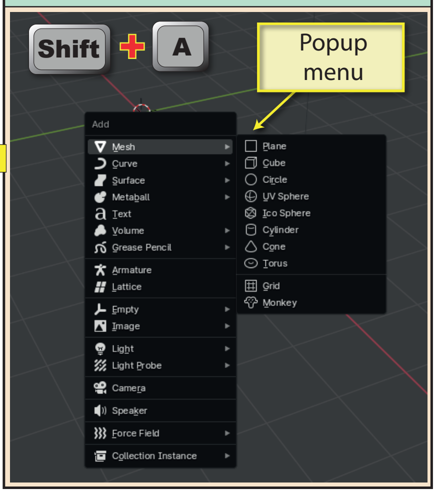
To add a new primitive to our scene, we need to use a menu option.

There are two ways to do this.

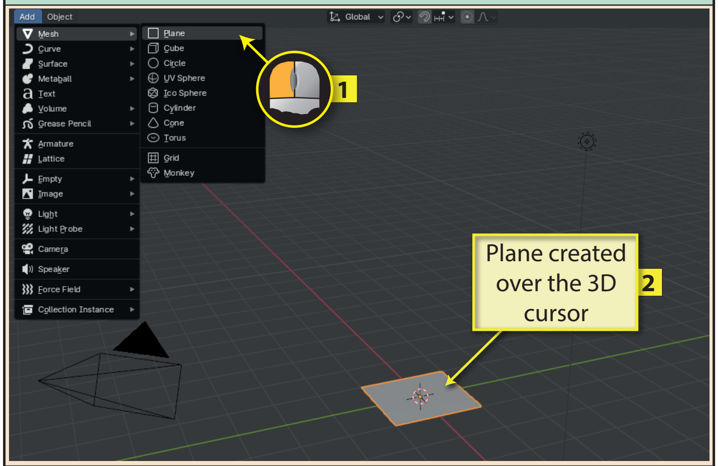
The first option is to choose **Add>Mesh** from the ***Viewport's*** main menu and then choose from the submenu list of primitives.



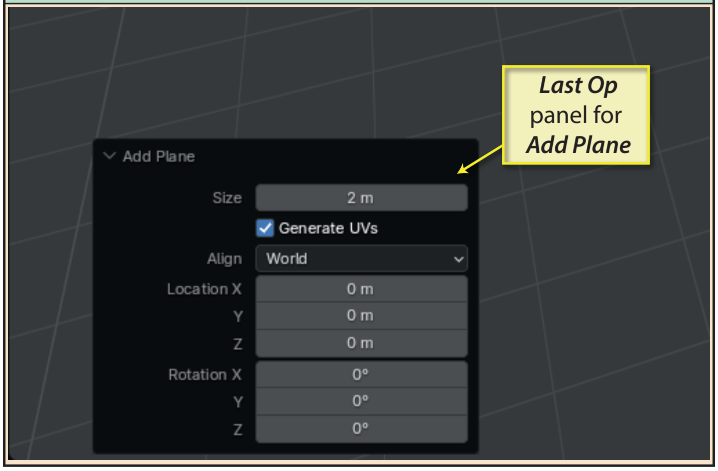
The second option is to choose **Shift A** to create the same menu but this time as a popup.



We'll start by creating a **Plane**, the first of the listed primitives and also the simplest with only four vertices, four edges and a single face. Remember, a new object is always placed over the ***3D cursor***.



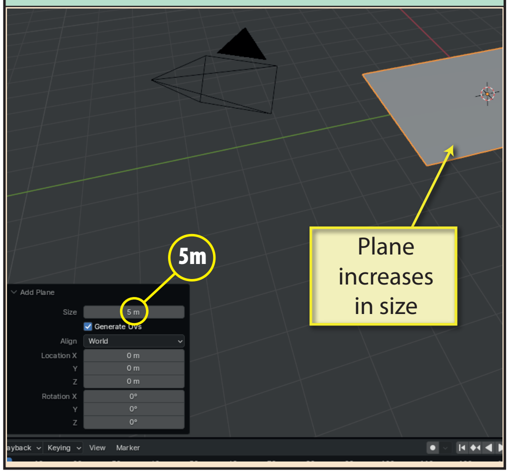
Many operations we perform during modelling creates a **Last Op** panel (also known as the ***Operator*** panel) in the bottom left of the ***3D Viewport***. This allows us to adjust various parameters of the operation we've just performed.



NOTE

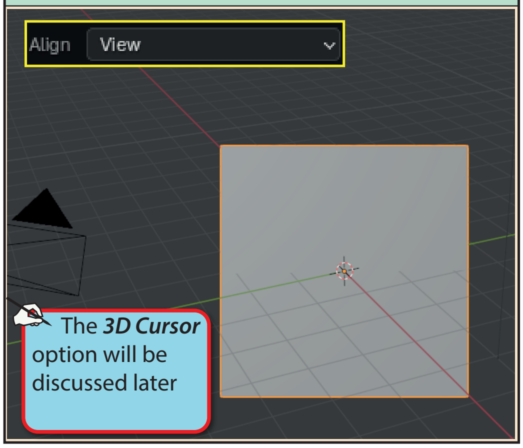
A Last Op panel only exists until another operation is performed at which point it is replaced by a new Last Op panel.

The first parameter for the ***Plane*** is **Size** which specifies the length of each side. If we change the value here, the ***Plane*** itself will change size.

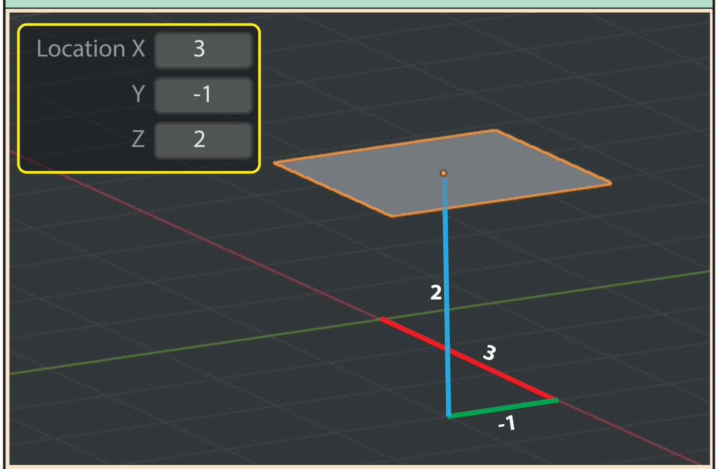


Generate UVs refers to texturing and will be discussed in a later chapter.

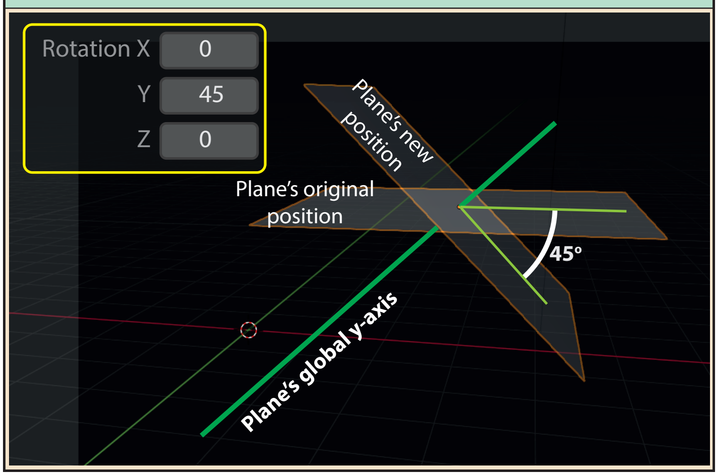
Align determines which ***z-axis*** the mesh is aligned along. By default it is the ***World's z-axis***, but if we change this to ***View*** (whose **z-axis** points out of the screen), the Plane aligns directly with our viewpoint.



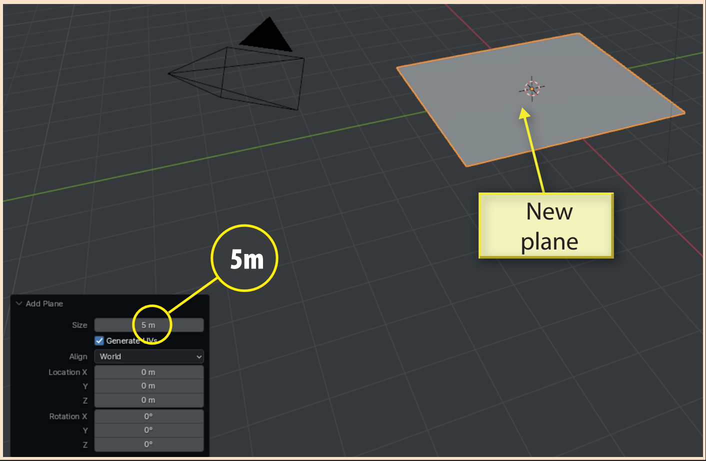
The next set of values, labelled **Location**, sets the position of the plane by moving it so that its origin is at the specified location. In the example below the plane is moved so that its origin is at location (3,-1,2).



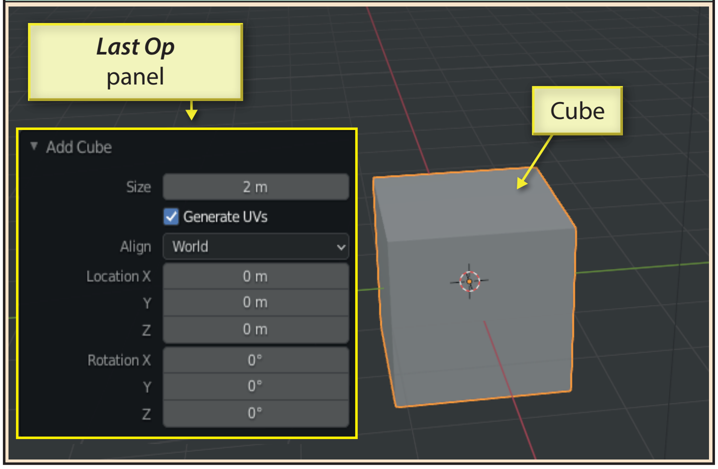
Finally, **Rotation** specifies the plane's rotation about the *x*, *y* and *z* axes. Normally, this will be measured from the plane's own global axes. **Rotation** is measured in degrees by default. Below a plane is rotated 45° about its *y*-axis.



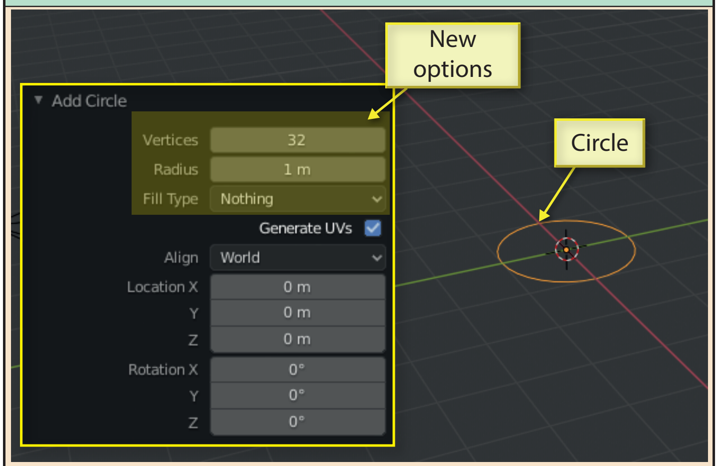
If we delete our plane and then create a new plane we'll see that Blender has remembered the last **Size** setting and has created this new plane with sides 5 metres in length but the ***Location*** and ***Rotation*** values are reset.



Once we've selected ***Add>Mesh>Cube*** from the ***3D Viewport***'s menu, we can see that the cube offers the same initial properties in the **Last Op** panel as the Plane:
Size, Generate UVs, Align, Location and ***Rotation***.

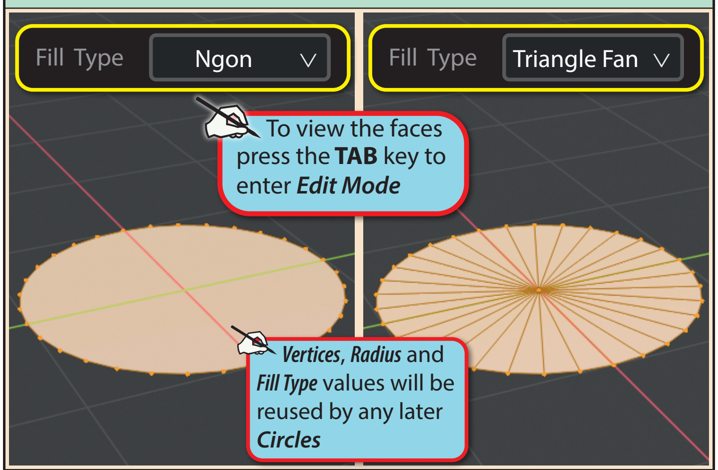


After deleting the Cube we can add the next mesh option, **Circle**. This creates a shape made only of vertices and edges. There are no faces. But the **Last Op** panel has some additional options.

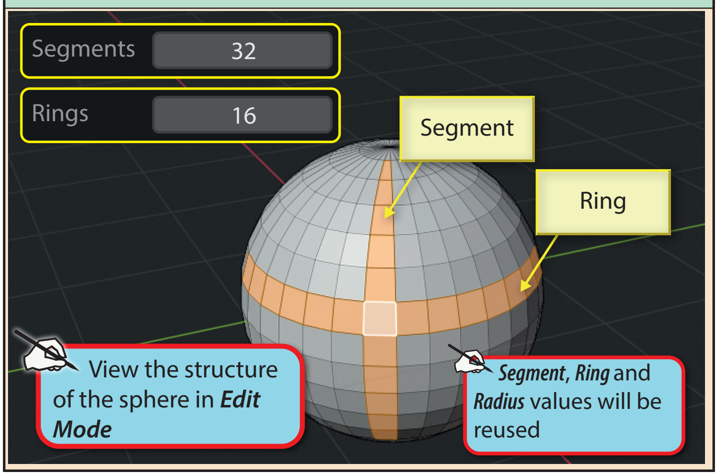


Vertices gives the number of vertices around the circumfrence of the circle. If we reduce this value the shape becomes less circle-like.

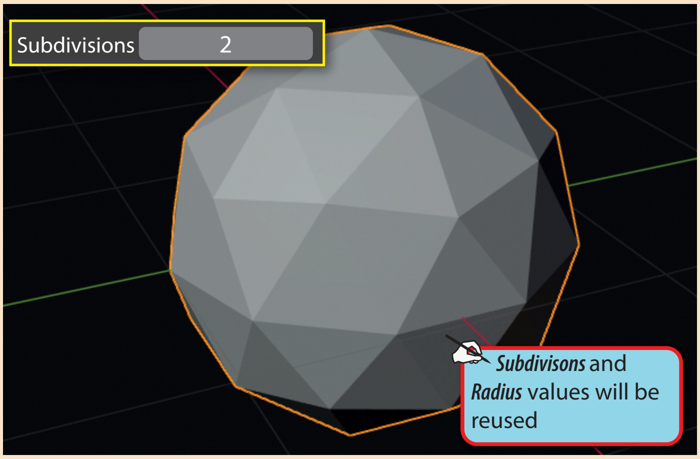
Radius sets the radius of the circle. **Fill Type** offers options to create one or more faces for the inner part of the circle. ***Ngon*** fills the circle with a single face. ***Triangle Fans*** creates a set of ***tris*** (3 edge faces) meeting at the circle's centre.



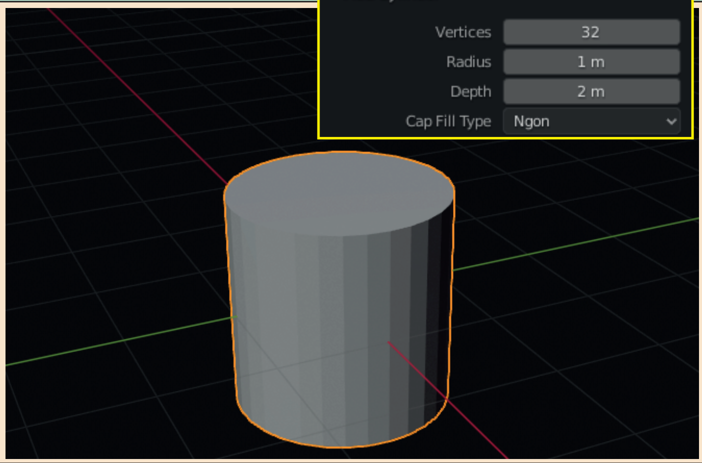
UV Sphere is the next mesh option. The ***Last Op*** panel has two new options. These are **Segments** and **Rings**. The faces that make up a single vertical loop is a ***segment***. Faces that make up a single horizontal loop is a ***ring***.



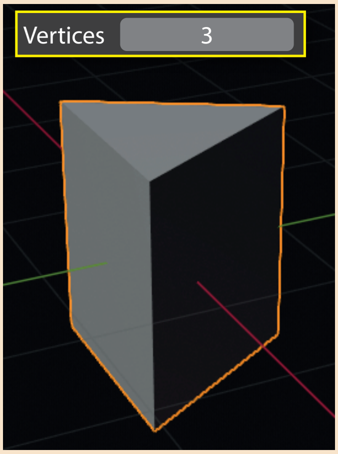
The **Ico Sphere** mesh is constructed from tris. The only new ***Last Op*** panel option is **Subdivisions** which, in effect controls how many ***tris*** are used to create the sphere.

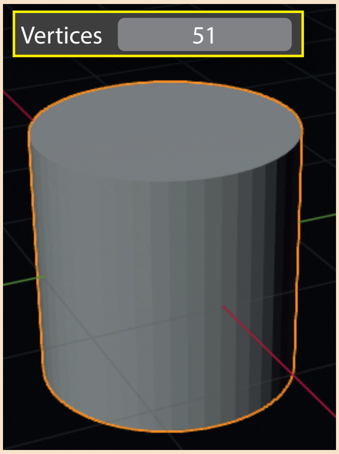


The **Cylinder** has some features similar to the *Circle* since the top and bottom of the cylinder are, in effect circles. This means that some of the ***Last Op*** panel options are similar to those of the ***Circle***.

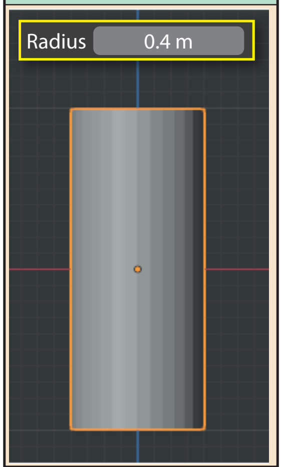


Reducing the **Vertices** value gives a less rounded shape. Increasing the **Vertices** makes the curve smoother.

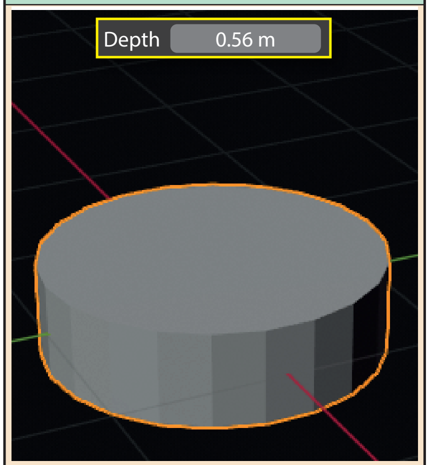




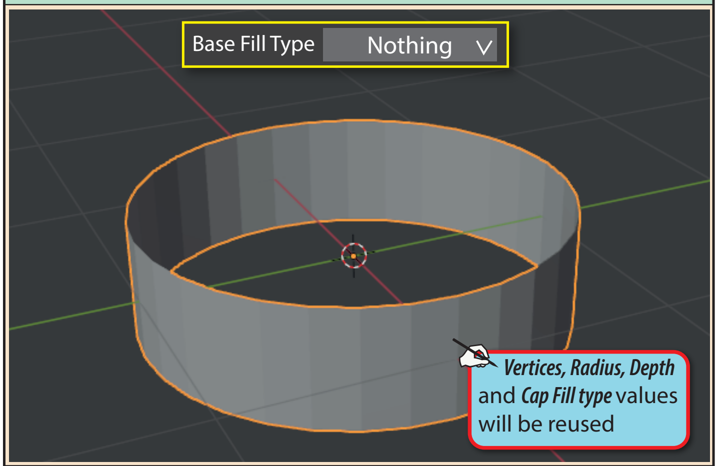
Radius sets the radius of the cylinder.



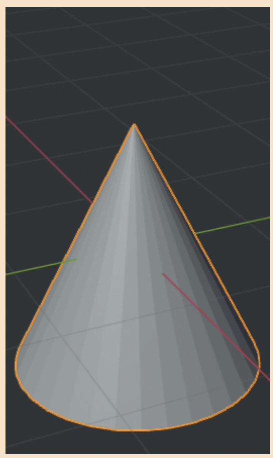
The **Depth** setting adjusts the height of the cylinder.



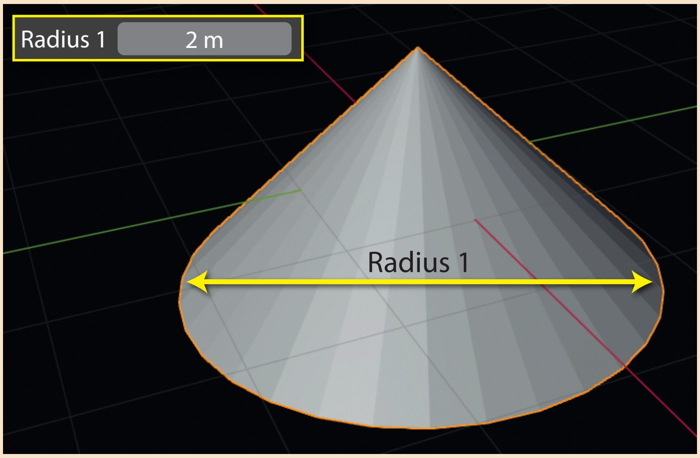
Cap Fill Type determines the type of face used to fill the top and bottom of the cylinder (***ngon*** or ***tris***) or to leave them unfilled.



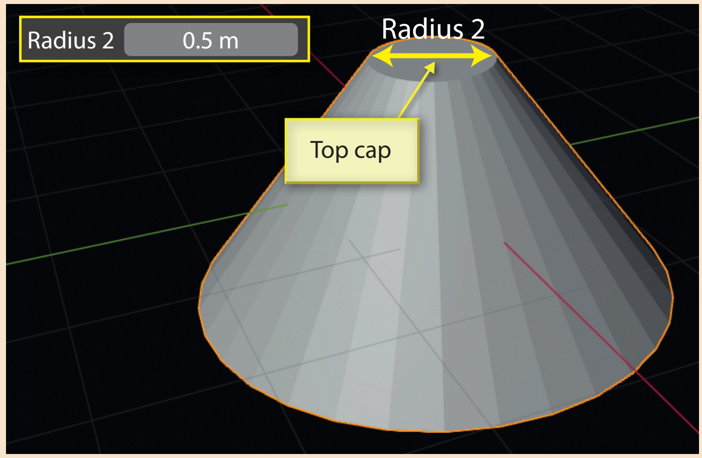
The **Cone** mesh's options are mostly familar but they include **two Radius values**.



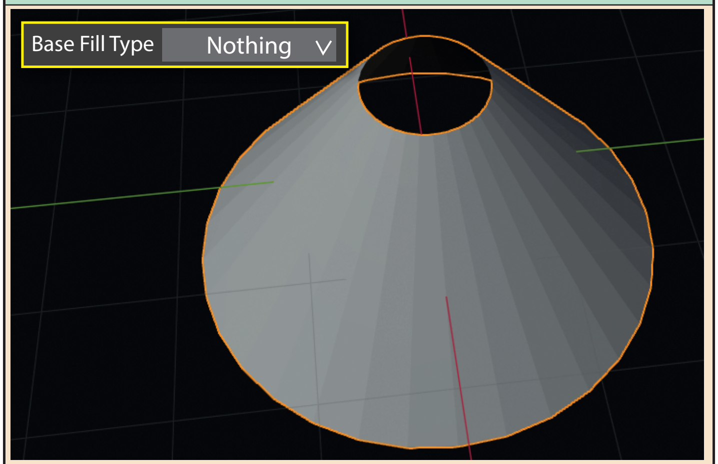
Radius 1 adjusts the width of the Cone's base.



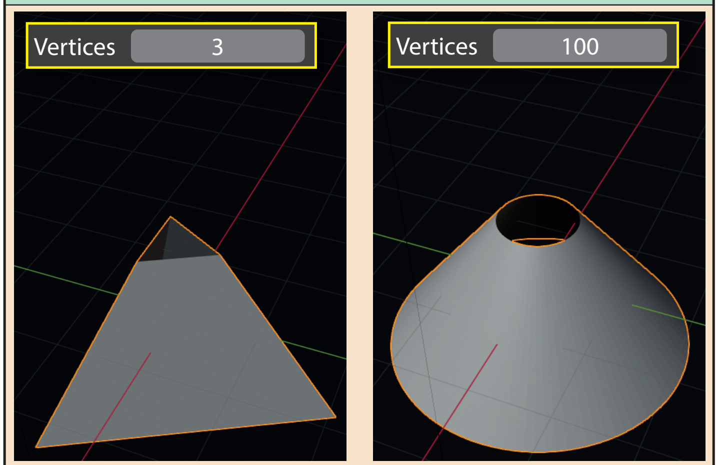
Radius 2 adjusts the width at the top of the Cone. For any value other than zero, we no longer have a true cone shape with the mesh acquiring a top cap.



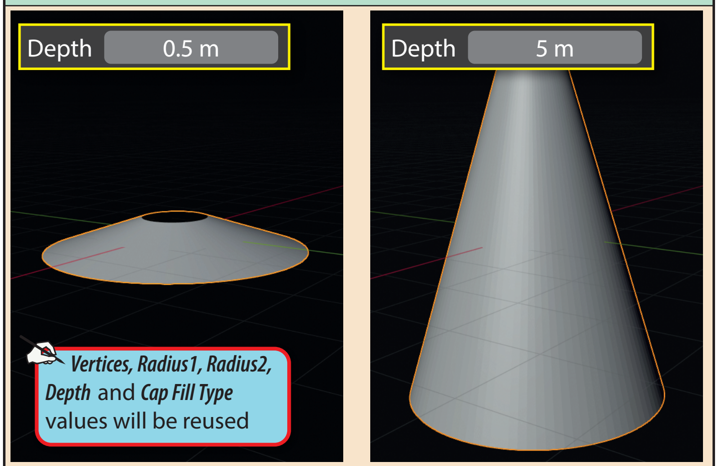
Base Fill Type determines how the base and top caps are handled (options being: *none*, *ngon*, or *tris*). Below, we see the result of the caps having been removed.



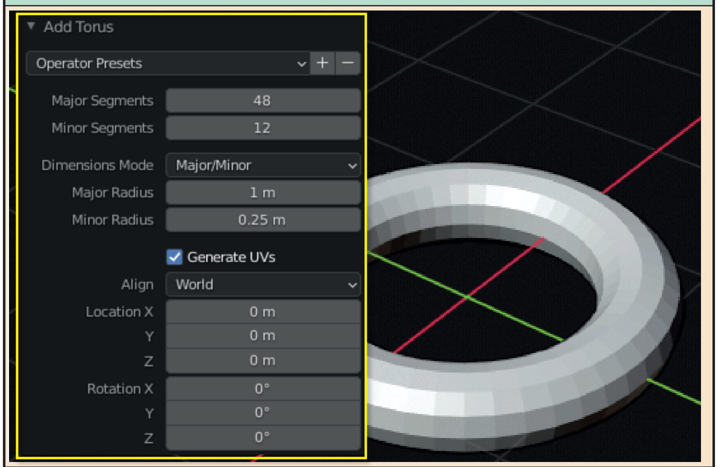
Vertices allow us to create a pyramid shape (values 3 or 4) or a very smooth curved cone shape (value 100).



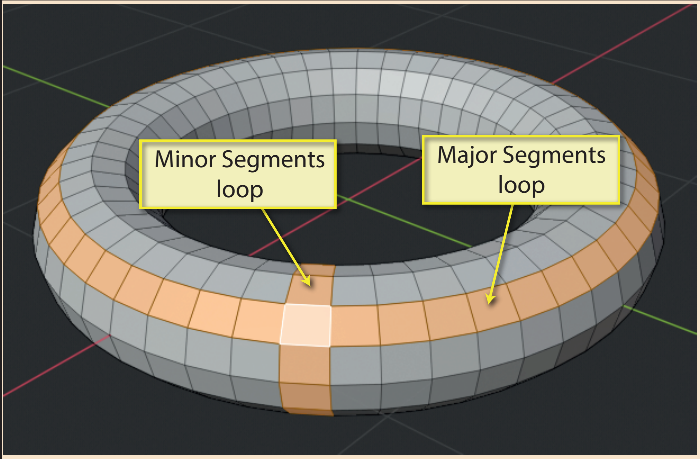
Depth sets the height of the Cone.



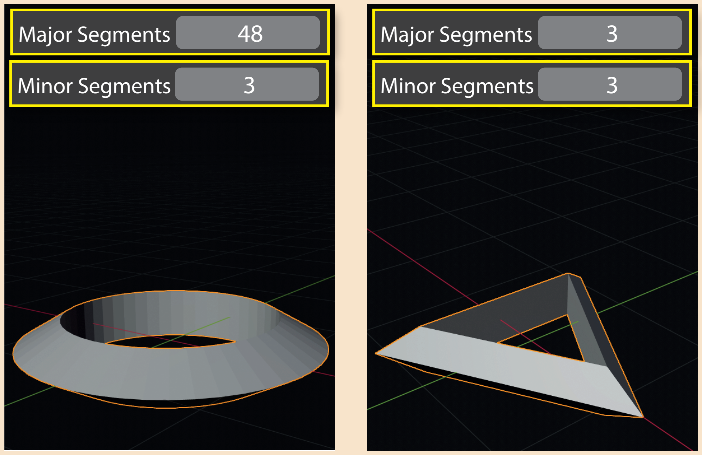
The **Torus** is a doughnut-shaped mesh with many unique options.



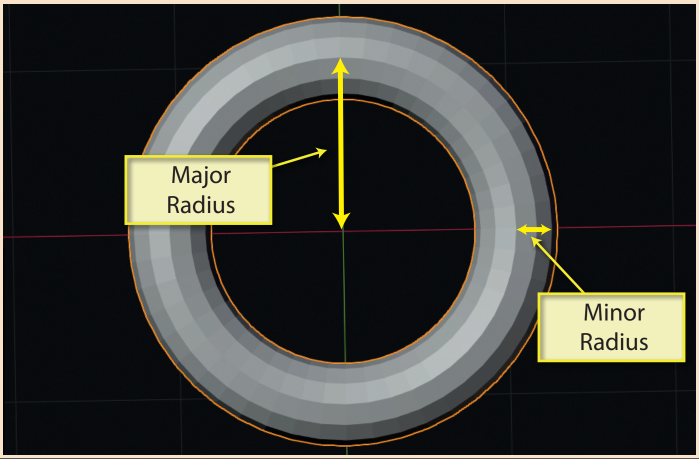
The option **Major Segments** refers to the number of faces that make up a horizontal loop. **Minor Segments** refers to the number of faces that make up a vertical loop.



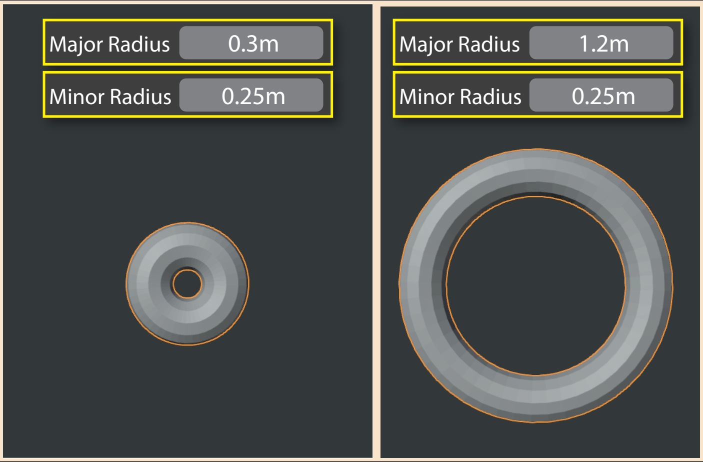
Reducing the **Minor Segments** to 3 gives us the shape shown below left. Reducing the **Major Segments** to 3 as well, gives us the shape shown below right.



The **Major Radius** is the distance from the centre of the Torus to half way through the solid ring.
The **Minor Radius** is half the width of the outer ring.

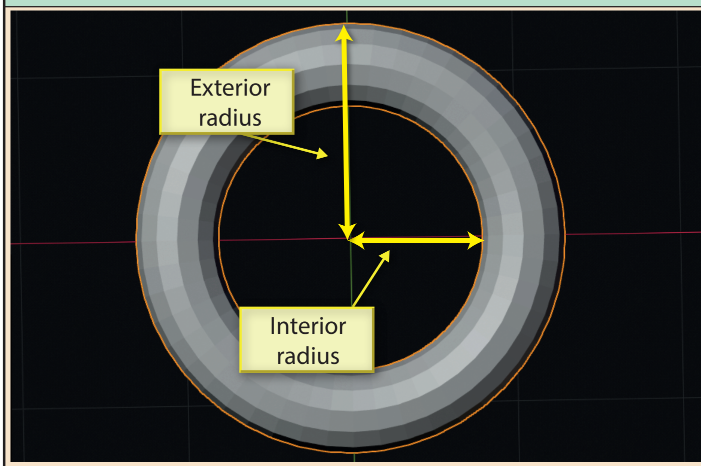


By adjusting the **Major** and **Minor Radius** values we change the overall size of the Torus, the size of the hole in the middle and the thickness of the Torus.



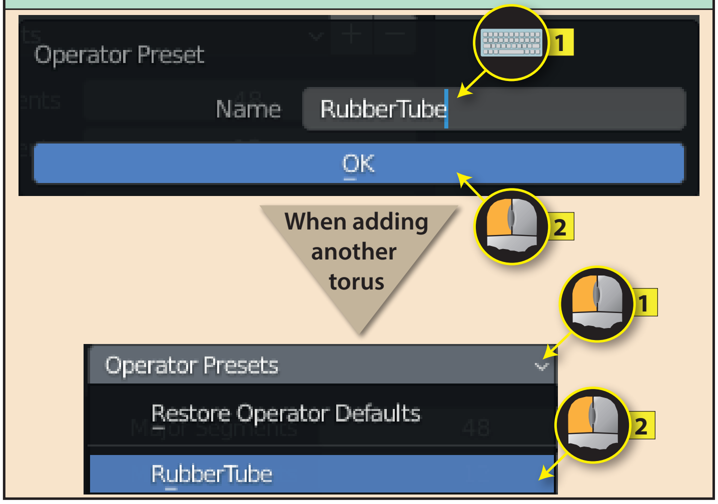
An alternative method of resizing the Torus is to select the **Dimensions Mode's Exterior/Interior** option This changes the two values displayed below to **Exterior Radius** and the **Interior Radius**.

The radii measured by **Exterior Radius** and **Interior Radius** are shown below.

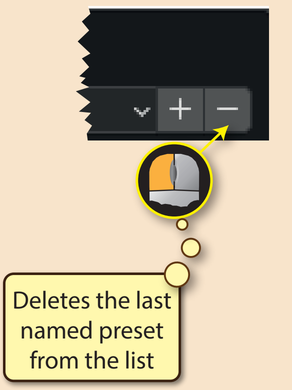


Because of the many options available when setting up a Torus, Blender offers a way of saving and naming a set of attribute values so that another matching Torus is easily created. To do this we must first set all the necessary Torus values and then click on the + sign to the right of **Operator Presets** in the *Command Settings panel*.

This opens a new panel where we can specify a name for the current value settings. When another Torus is added, the name can then be selected to apply the associated settings to the Torus.



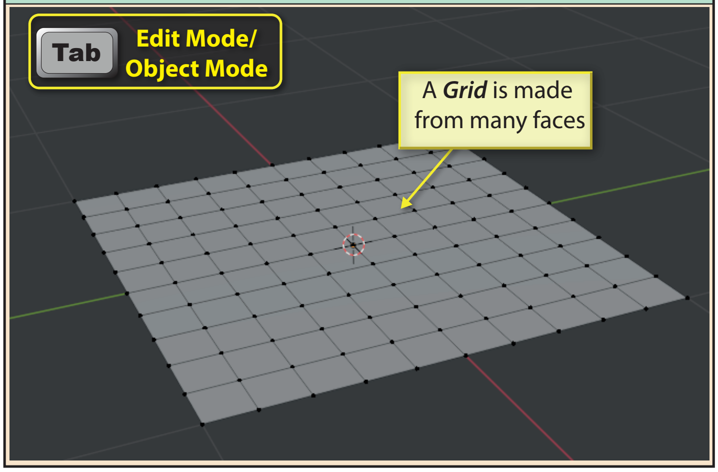
Clicking on the - icon will delete the last named preset to have been used.



Even if we haven't saved the various Torus settings in the way described, any new Torus will make use of the current settings of the following parameters:

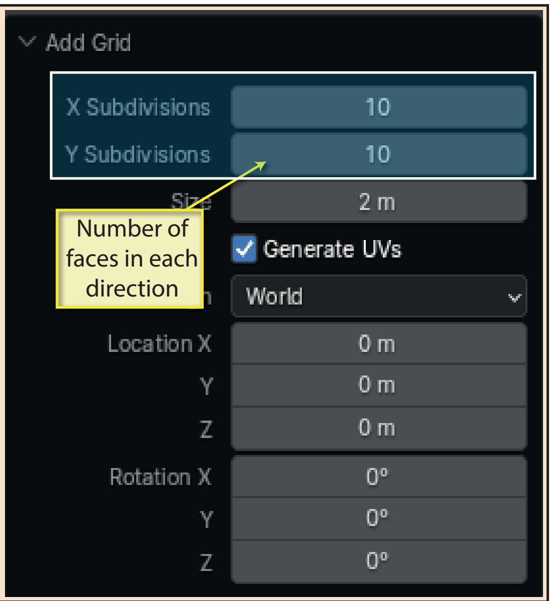
Major Segments
Minor Segments
Dimensions Mode
Radius settings

The next mesh option, **Grid**, may look identical to the ***Plane*** mesh but if we look at its structure in ***Edit Mode***, we can see that it is constructed from many more faces.

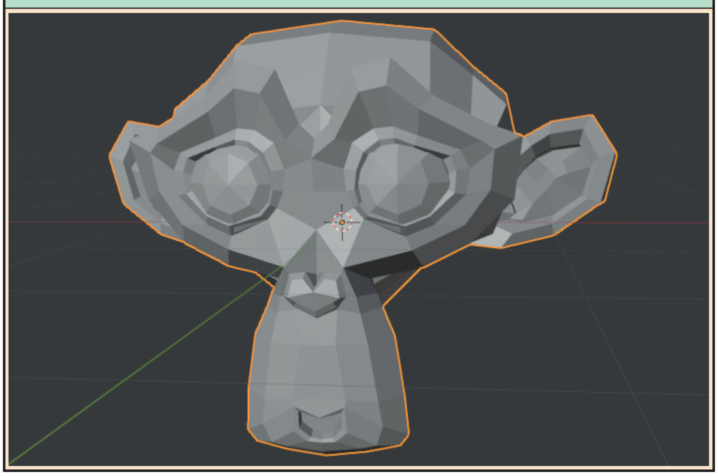


Grid's *Last Op* panel options allow us to specify the number of faces in both the *x* and *y* directions.

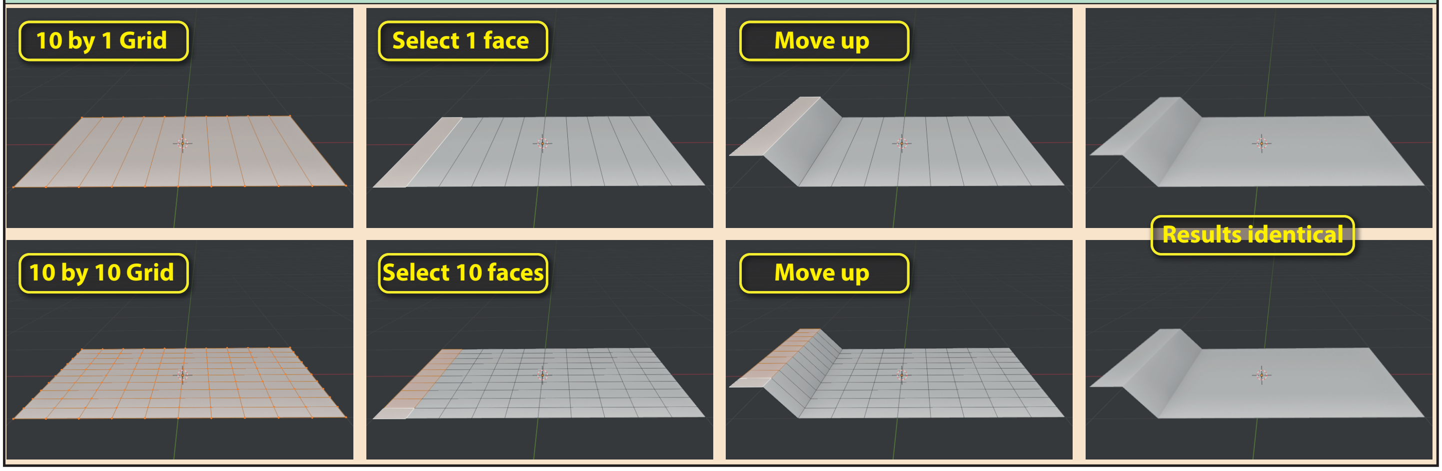
X Subdivisions, *Y Subdivisions* and *Size* settings are reused by subsequent ***Grids***.



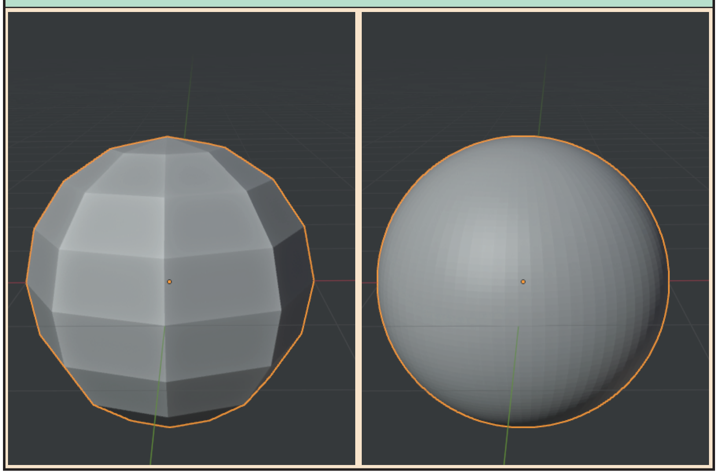
The final mesh is a monkey head - affectionately known as **Suzanne.** Although not a true primitive, it is often used to show off various features of Blender.



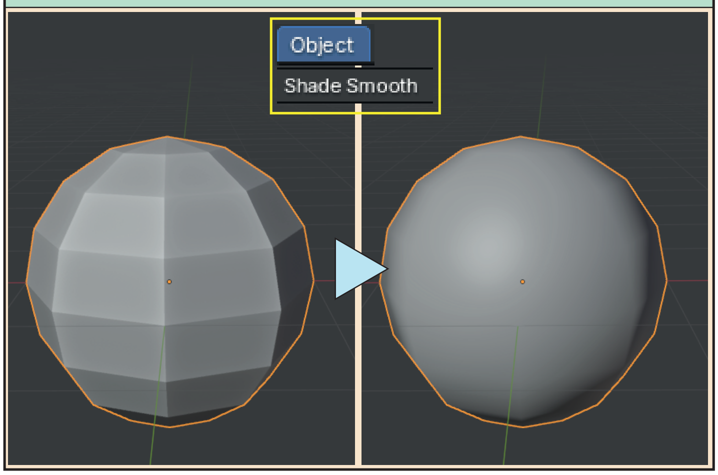
When creating a mesh we should always go for the minimum number of faces we require since not only will we reduce memory and processing requirements, but this can also make the modelling process much easier when we are working in ***Edit Mode***. For example, one of the Grids below is created with 10 faces while the other has 100. In the modelling process we want to raise one end of the grid. Both end up with exactly same result but one takes a lot less effort for both the modeller and the machine.



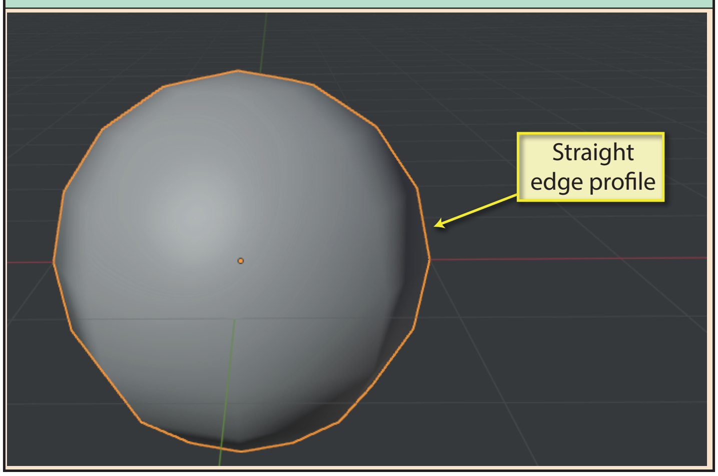
One situation where we might be tempted to increase the number of faces is when creating a curved surface. For example, we can see that the second sphere below looks more curved than the first.



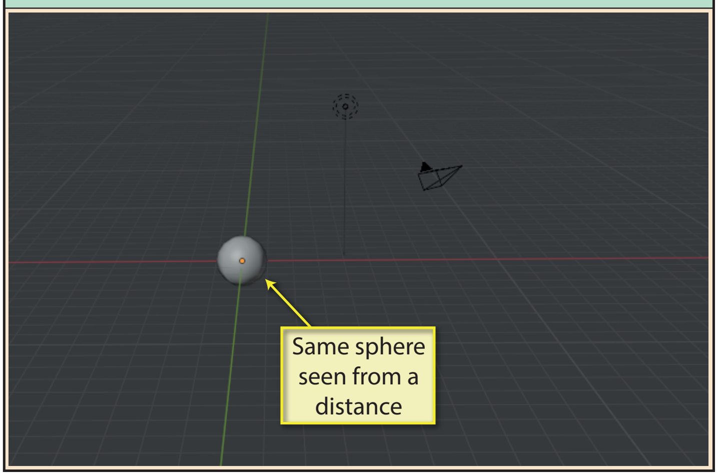
But if we use the first version of the sphere and apply smooth shading (***Object>Shade Smooth***) we get a more curved look to the object without adding more faces. This effect is achieved by adjusting the normals of the sphere.



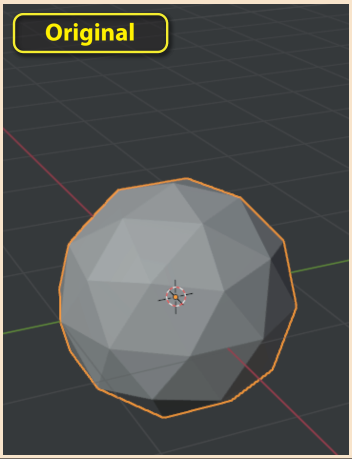
In fact, the only visual clue to the sphere's low face polycount is the straight edges on its profile.

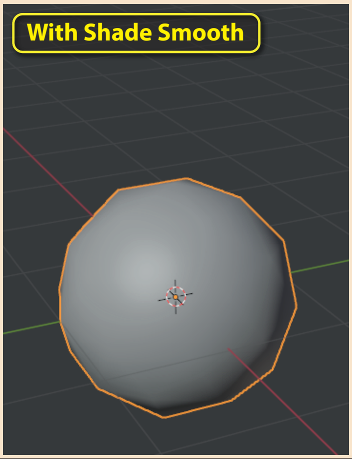


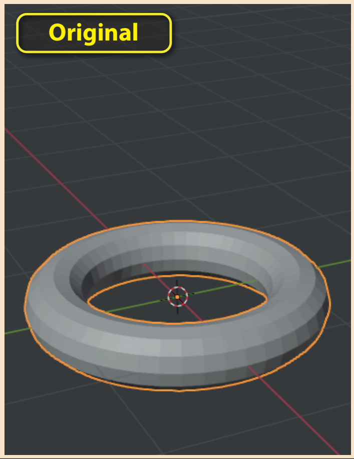
However, if the sphere will only ever be viewed from a distance in our completed scene, the problem with the the profile disappears as seen below.

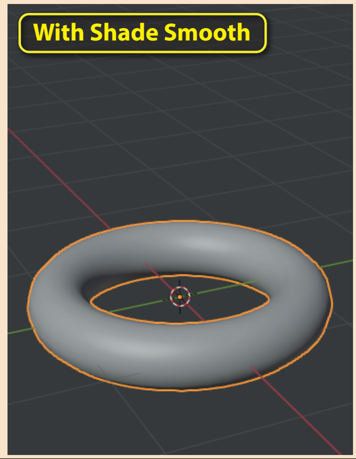


The ***Shade Smooth*** option works well with the ***IcoSphere*** and ***Torus*** as we can see from the before and after pictures shown below.

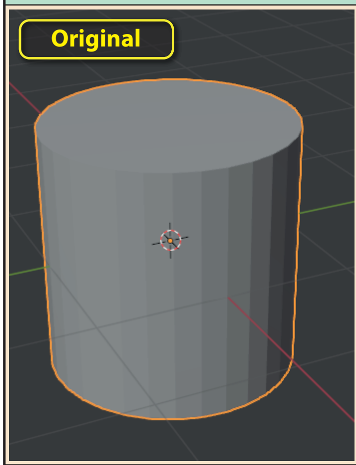


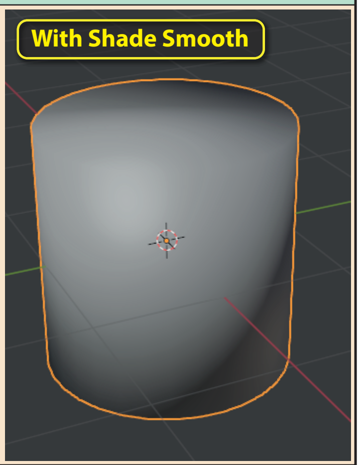




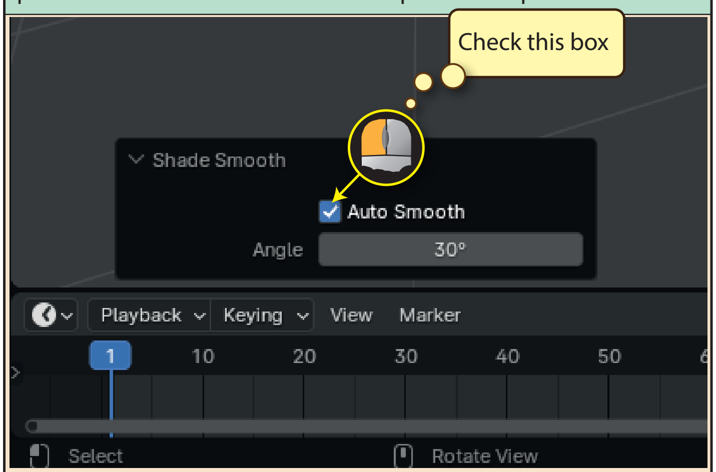


However, when we apply ***Shade Smooth*** to a ***Cylinder***, the result looks wrong.

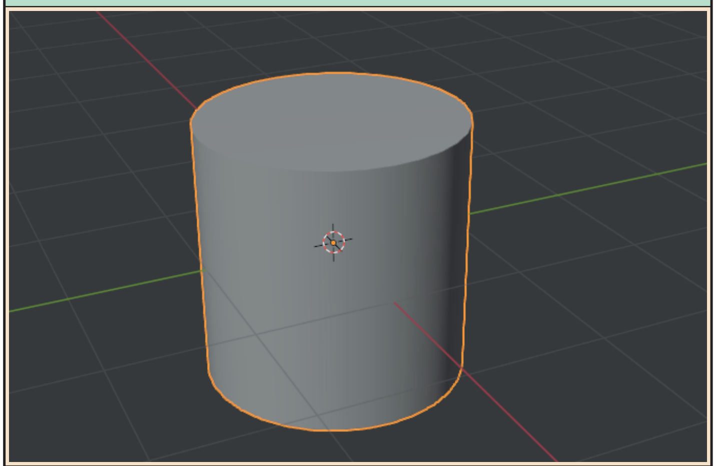




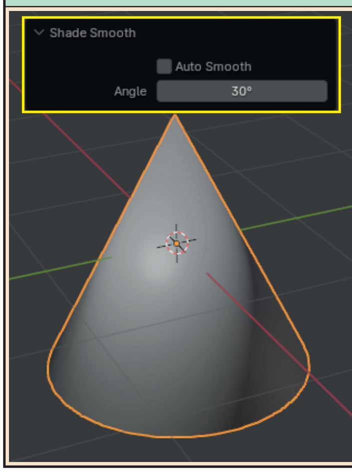
The trouble is that ***Shade Smooth*** is attempting to smooth out the whole surface of the Cylinder when all we want is to smooth out the vertical, curved section. Luckily, the ***Last Op Panel*** has a parameter for ***Shade Smooth*** that helps with the problem.

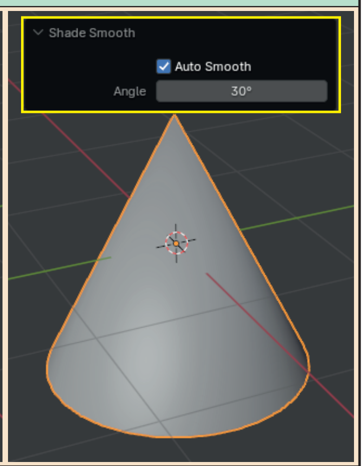


By checking **Auto Smooth**, Blender only smoothes out faces which are at an angle of 30° or less to each other - the angle value can be changed. Since the top faces of the cylinder are at an angle greater than 30° to the side faces, we can achieve a better result.



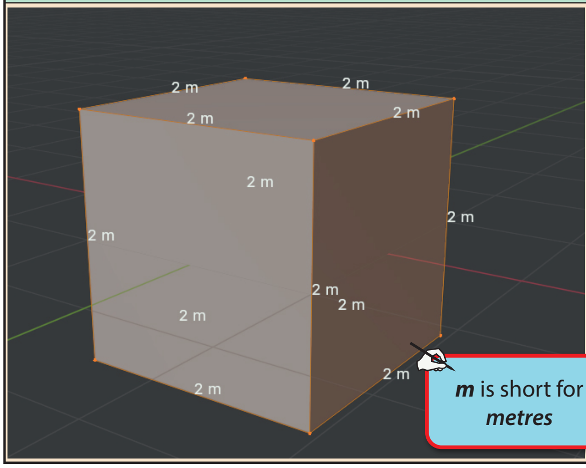
We need to check the same ***Auto Smooth*** option when applying ***Shade Smooth*** to a ***Cone***.





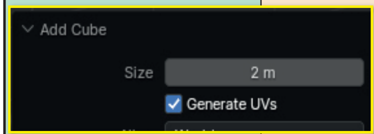
Measurement Units

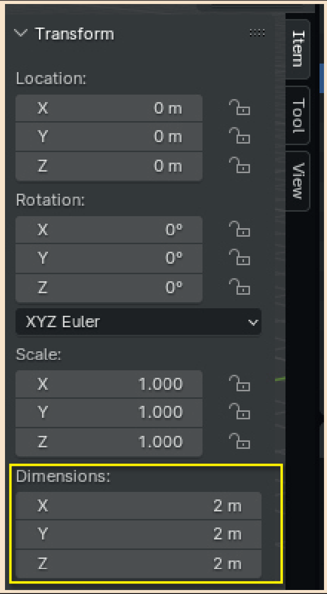
When we start a new project in Blender, the Metric system is used by default, with one Blender unit of distance set equivalent to one metre. This means that our default Cube is 2 metres in all three directions.



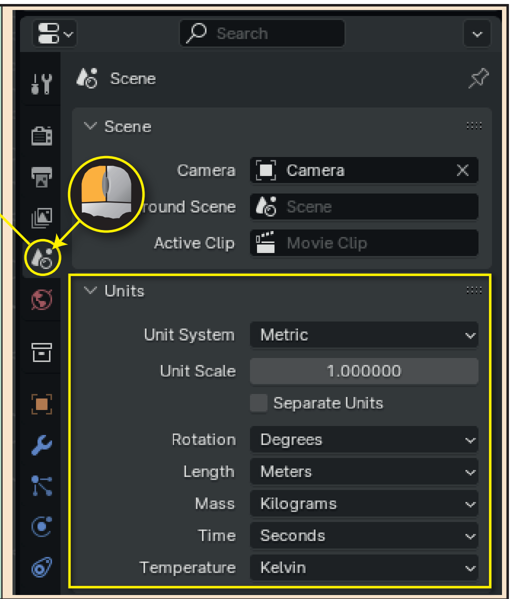
m is short for ***metres***

We can see the size of an object in both the ***Last Op*** panel when the object is created, and also in the ***Sidebar's Item*** page where the size in each of the three dimensions is given.

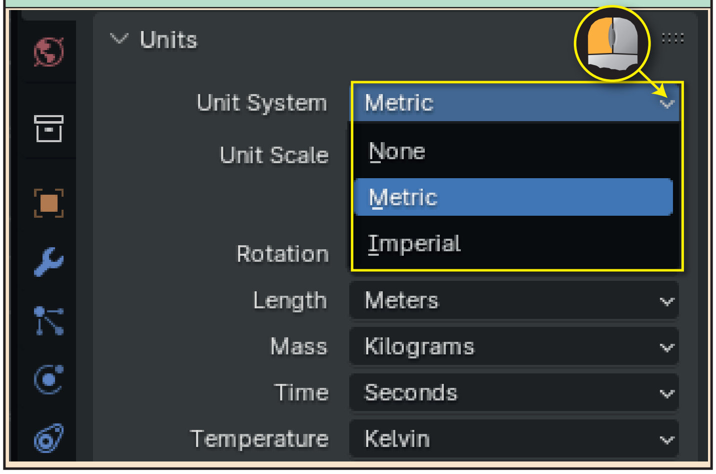




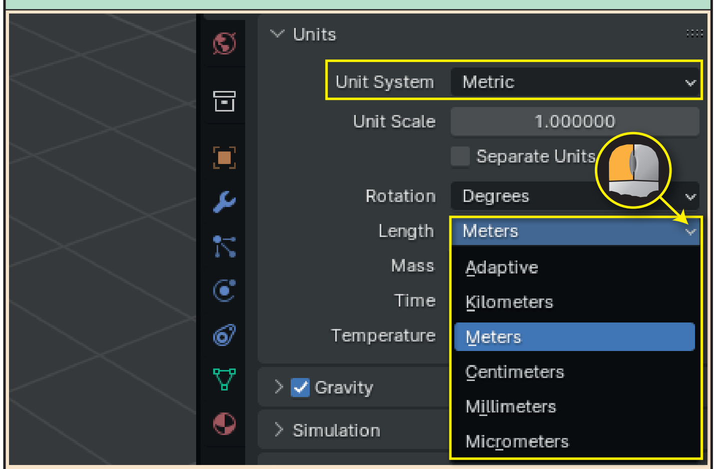
The units we are using can be set in the **Scene** page of the ***Properties Editor***.



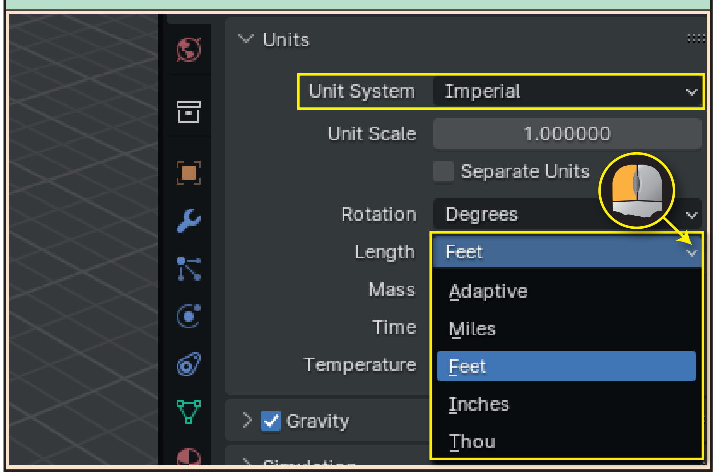
Unit System gives us a choice of ***Metric*** or ***Imperial*** with a third option, ***None***, removing any link between real-world size and Blender measurement units.



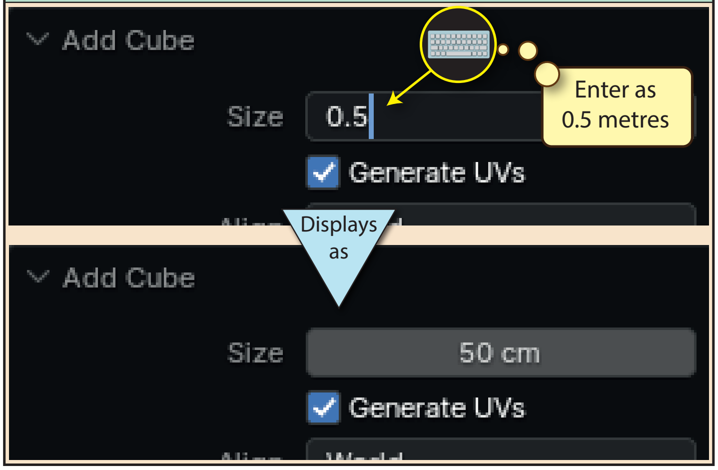
When we choose ***Metric***, the **Length** field lower down the list of fields offers measurement units of anything from a kilometre down to a micrometre.



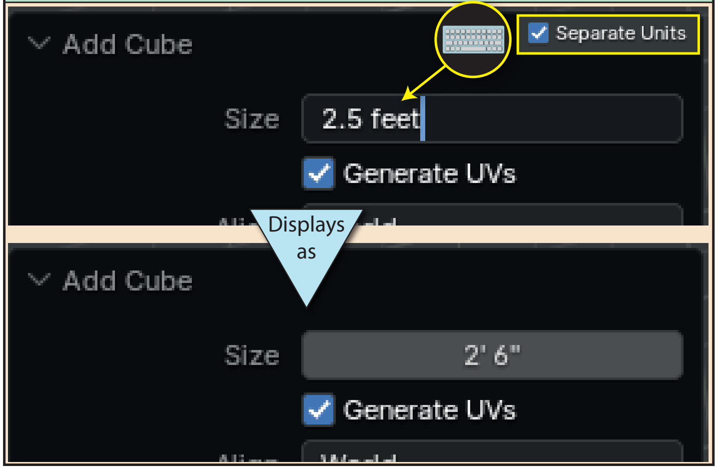
When we choose ***Imperial***, we get the old British measurements which are still used in the USA. This ranges from miles to thou (one thousandth of an inch).



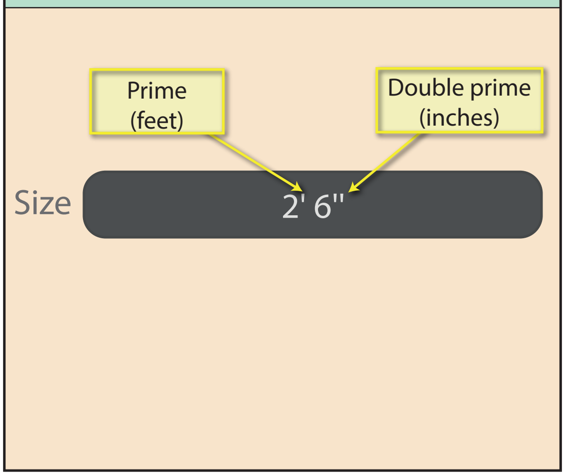
Both ***Metric*** and ***Imperial*** have a ***Length*** option labelled **Adaptive**. This allows Blender to use the most appropriate units when displaying values. For example, if we are working in Metric and create a Cube defining it size as 0.5 metres, Blender displays 50cm.



Separate Units is a checkbox, which, when selected, shows named units rather than fractions where possible. Here, when working in ***Imperial***, we have created a Cube which has a size of 2.5 feet this is then displayed as 2 feet 6 inches.

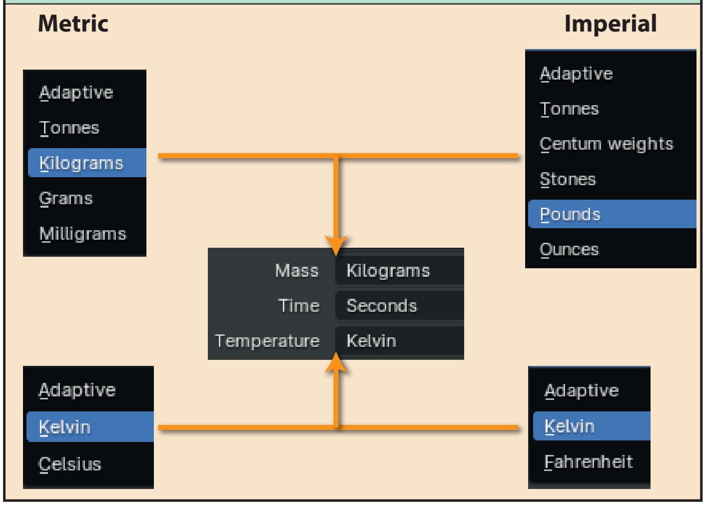


If you are not familar with the Imperial system, the tick marks by the numbers - properly called "prime" and "double prime" - are the shorthand used for "feet" and "inches".

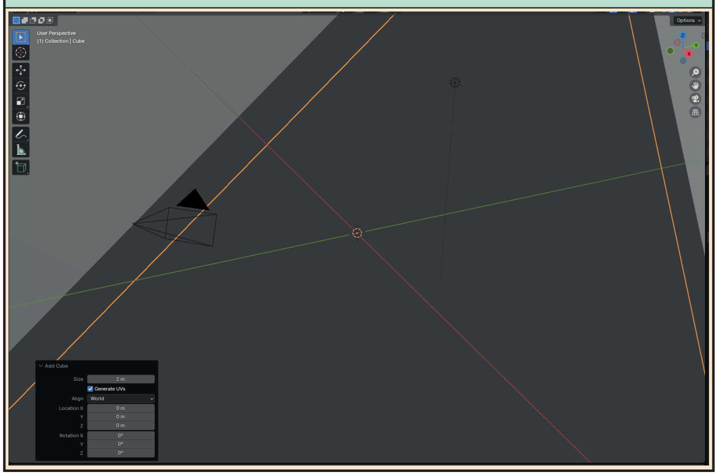


Rotation can be set to ***degrees*** or ***radians***.

Mass, **Time**, and **Temperature** are only of use when we are using physics to create realistic animations. But note that ***Mass*** and ***Temperature*** have different settings for ***Metric*** and ***Imperial***.



Unit Scale should scale distances appropriately. But, making use of it causes display problems as shown below when a 2 metre Cube is created. So, it is best to ignore this field, leaving it set to 1.0.



However, ***Scale*** in the ***Viewport Overlays*** panel can be adjusted when working on a different scale from the default metres or feet. For example, setting ***Scale*** to ***0.001*** when using metric, will set the smallest grids shown in named and user views to exactly 1mm.