# 3D modeling using Blender



Ciril Bohak - 2022 -





# Lecture 01 User interface

# Overview

- 1. User interface
- 2. Scene navigation
- 3. 3D cursor
- 4. Objects in scene

# User interface

Menijska vrstica



Urejevalnik lastnosti

The default user interface consists of (look the image above):

- Topbar (blue) contains basic commands and information, which are always awailable to the user (working with files, rendering, UI control, help, UI selection, scene control, rendering options, and basic information about the scene and objects.;
- 3D view (pink), shows curren scene and editing tools;





- Timeline (green), intended for animation, enables playing, editing, viewing, keyframe overview etc.;
- Outliner (red), showing what is in the current scene and what is the hierarchical relationship between objects in the scene, shows the basic properties of objects such as: visibility, the ability to interact with the object and the ability to depict objects;
- Properties (yellow), which is intended to edit the properties of individual selected objects, global properties of rendering, properties of materials, layers in the scene, properties of the scene, the world, etc.

The user interface is extremely flexible and is divided into different parts using components such as: screens, areas, regions, tabs and panels, and controls (buttons and menus). The interface can be completely customized by the individual. You can divide it into several windows and show the desired things in each. Basically, Blender offers a set of several interface adaptations that are adapted to the specific application: default, full 3D view, animation, composition, scripting, video editing... You can switch between them in the information editor.

### Scene navigation

Blender offers several options for navigating the scene displayed in 3D. You can use both a mouse and a keyboard (you can also use a so-called 3D mouse or other input device). Navigation is similar in most other editors (eg when editing texture mapping, defining animation curves). In the following, the keys will be marked with square brackets (e.g. [a], [enter], [space]). We will also label mouse buttons and the wheel ([LMB], [MMB], [RMB] and [MW]) in a similar way.

### View turning

We turn the view in the scene with the middle mouse button [MMB]. Hold the middle mouse button over the scene and with mouse movements turn the view of the scene as if you were moving in orbit above the center of the object.

Rotating the scene around the horizontal and vertical axes can be achieved with the keyboard, where you can use the numeric keypad [num 2], [num 4], [num 6], [num 8]. You can also use the numeric keypad to quickly switch between the default aligned views:

- Default camera view [num 0],
- aligns the active camera with the current view [ctrl] + [alt] + [num 0],
- front view [num 1] and back view [ctrl] + [num 1],
- right [num 3] and left [ctrl] + [num 3],
- Top [num 7] and bottom [ctrl] + [num 7] or [num 9].

### View panning

The view in the scene can be moved up, down, left and right with the mouse or keyboard. When using the mouse, use the combination [shift] + [MMB] to move with the keyboard: left [ctrl] + [num 4], right [ctrl] + [num 6], up [ctrl] + [num 8] and down [ctrl] + [num 2].





### Zoom in/out

You can zoom in or out using the keyboard or mouse. When using the mouse, use the mouse wheel [MW] for continuous zooming in/out and the combination [ctrl] + [MMB]. When using the keyboard, use [num +] to zoom in and [num -] to zoom out.

### Zoom with window

You can use the [shift] + [b] key combination to zoom in on the selected part of the scene determined by the selection window. This feature brings us closer to the selected part of the view over the entire 3D view.

### Fly and walk in through the scene

The blender allows us to move around our scene even by walking and flying functions. While in scene mode we move as if we were piloting an airplane, while walking around the scene we move as if we were walking around the world - we are limited by the ground and gravity acts on us.

### Fly mode

Switch to flight mode with the key combination [shift] + [~] (left of key 1). In this mode, the camera through which we watch the scene is rotated by moving the mouse. We move around the world with the keys that are mostly used in computer games: forward [w], left [a], back [s], right [d], down [q] and up [e]. The speed of moving around the scene can be changed using [MW]. We can also use t. i. a teleport that moves us to the first surface we see in front of us. Use the [space] key to activate the teleport. When we are satisfied with the shift of the view, we can confirm it with the [enter] key or by pressing [LMB]. Use the [esc] or [RMB] key to reset the view to the position before entering flight mode.

### Walk mode

We switch to walking mode in the same way as to flying mode. When we are in flight mode, we can turn on gravity with the [tab] key. We can then move around the scene in the same way as in first-person games. The shortcuts are the same as in flight mode. In walking mode, you can also jump, which you do using the [v] key. Other shortcuts and promotions are the same as in flight mode.

### Switching between the projections

We can look at the world with a parallel projection or with a perspective projection. Switch between them with the [num 5] key.

### Focus view

During 3D modeling, we often want to focus on making individual details. Other objects in the scene can make this difficult for us. Blender allows us to display only the active object and hide the rest with focused gaze. Switch between normal and focused view with the [num /] key.

### Four simultaneous views

Many times we want to look at the scene from several angles at the same time. Mostly in three aligned views and in perspective. Such a method is called "Quad view". In the 3D viewport, switch to this view with the combination [ctrl] + [alt] + [q]. We get out of it in the same way. In this view, we can manage each part of the window independently and set the view that suits us best.





### 3D cursor

A 3D cursor is a point in 3D space that can be used for many purposes. We will also learn about some such scenarios during the workshop. In the following, we will present how to operate a 3D cursor. The 3D cursor represents a red-white circle with a black cross.

### Locating

The easiest way to place a 3D cursor in space is to use the mouse. In the view, click [LMB] on the selected position and the cursor will move there. For more precise positioning in 3D space, it makes sense to define the location of the cursor in two separate aligned views (e.g. left and front view).

The position of the 3D cursor can also be determined in the right 3D view editing panel, which is activated with the [n] key, where the "view" tab is selected, then numeric values along the individual axis can be entered for the cursor position.



Reset the cursor position with the combination [shift] + [c].

# Objects in the scene

The scene in Blender consists of one or more objects. These can be: lights to illuminate the scene, 2D or 3D shapes that define the models, skeletons for their animation and / or cameras to depict the scene.

Each object in Blender consists of two parts: (1) object and (2) object data. The first stores information about the position, orientation and size of the object in the scene, while the second stores information about the type of object (e.g. grid geometry in the case of a 3D object, or camera properties in the case of a camera object).

### Object interaction types

Blender distinguishes several ways of interacting with objects from which you can choose at the bottom of the 3D view bar. The ways of interaction are:

- Object mode is the default mode of interaction that allows interaction with objects as a whole (e.g., moving, rotating, enlarging objects);
- Edit mode the mode is intended for all readable objects and allows changing the shape of objects (e.g., in the case of mesh geometry changing the corners, edges, surfaces; in the case of curves, changing the control points);
- Sculpt mode is intended for mesh objects for the use of sculptural tools;
- Vertex paint mode is intended for coloring corners in mesh geometry;
- Weight paint mode is designed to determine the weight of the corners in the mesh geometry;

#### 5 | Ciril Bohak @ FRI | 3D modeling using Blender





- Texture paint mode is designed to draw texture directly on a 3D model of mesh geometry;
- Particle edit mode is only available for mesh geometry and is intended for arranging particle systems e.g. hair;
- Pose mode is intended to regulate the distribution of bones within buildings and is available for working with bones;
- Edit strokes mode is intended only for drawing auxiliary sketches.

### Object pivot

The pivot point determines where in the scene the object is located. By transforming (moving, rotating or scaling) we can determine where and how the object is located in space. If necessary, the pivot point of the object can also be subsequently moved in the menu accessible with [RMB].

Shade Smooth		
Set <u>O</u> rigin	×	Geometry to Origin
🗐 Copy Objects		Origin to Geometry
R, Paste Objects		Origin to 3D Cursor
		Origin to Center of Mass (Surface)
Duplicate Linked		Origin to Center of Mass (Volume)
Rename Active Obj	ect F2	
Mirror		
Snap		
Parent		
Move <u>t</u> o Collection		
Incert Keyframe		

### Object types

### Mesh geometry

Mesh geometry is an object composed of polygons, edges and / or vertices and can be edited with the help of Blender's tools for editing mesh geometry, which will be discussed later in the workshop. The blender offers us to create grid primitives (cube, sphere, cylinder, torus, etc.) from which we can then further create the desired 3D model.

### Curve

Curves are mathematically defined 1D objects that can be edited with control levers. Thus we can regulate their curvature and length.

### Surface

Surfaces are mathematically defined 2D objects that, like curves, can be edited via control points. They are basically designed to create rounded shapes and organic landscapes.

### Metaballs

Metaballs are mathematically defined 3D objects that define the 3D volume where an object exists. We cannot change their shape through corners or checkpoints. Objects have the property that they blend smoothly when they get close enough.

### Text

The object creates a 2D representation of a string.

#### Bones or Armature

Armatures or bones are objects intended to be attached to other objects as a skeleton (rigging) for the purposes of animation.

### Lattice

Lattices are inexpressible mesh objects to achieve additional control over the shape of objects (most often in animation).





### Empty object

Empty objects are intended to store selected locations in 3D space to help achieve the desired results (e.g., to control the position and movement of other objects).

### Camera

A virtual camera that defines what will be rendered.

### Light

The lights are designed to define the lighting of the scene. There are several types of lights available with which we can achieve the desired illumination of the world.

### Force fields

Force fields are useful in physical simulation. They add external forces to the simulation and thus influence the movements of objects in 3D space.

#### Instance

When duplicating items, an instance of a group can point to an existing object or group of objects. When we change the look of the original object, the duplicate will also be adjusted. This also eliminates unnecessary duplication of all properties of duplicate objects.

### Common properties

All objects have common features: type, radius / size, position, orientation and possible alignment with the view.

### Object selection

Selecting or marking of objects determines over which objects the actions initiated by the user will be performed. Selection is available in object mode and edit mode. The blender distinguishes three states of the object (see figure): unmarked (bordered in black), highlighted (bordered in orange) and active (bordered in yellow).



The easiest way to mark an individual object is [LMB], and to mark / unmark additional objects by holding down the [shift] key.

The most useful ways of marking are:

- select with window (key [b]), where the box selects which objects you want to mark;
- select with brush (key [c]), where we use a "brush" in the shape of a circle to "paint" the individual we want to mark, and the size of the circle can be determined with [MW].
- lasso selection (combination [ctrl] + [LMB]), where we delete the objects we want to mark;
- (de)selecting everything (key [a]) marks or mark all objects;
- select more / less (combination [ctrl] + [num +] or [num -]) expands / shortens the designation to parents / descendants in the hierarchy.

There is an even wider range of labeling options but it goes beyond the scope of the lecure.





### Editing objects

We can perform various actions on the marked objects. The most commonly used object editing campaigns are:

- deleting marked objects ([x] or [delete] key), the deletion must be confirmed;
- hiding highlighted objects ([h] key), hide unmarked objects (combination [shift] + [h]), show all hidden objects (combination [alt] + [h]);
- joining objects (combination [ctrl] + [j]) merges the selected objects into the last selected object. Facilities must be of the same type.

### Transforming objects

The basic transformations of objects are grab or move, rotate, and scale. All three transformations are available to us both in object mode, where they work on the object as a whole, and in edit mode, where they act on the selected subdivision of the object.

### Translate

Start the scroll action with the [g] key. Use the mouse to take control of the object and move it to the desired position in the scene. The new position of the building can be monitored in the lower left part of the 3D view window. The action must end with [LMB] or [enter] or aborted with [RMB] or [esc].

### Rotate

Start the rotation action with the [r] key. Use the mouse to take control of the object and you can rotate it around the axis through the screen. By pressing the [r] + [r] key twice in succession, we switch to the "dolly" rotation mode, where the object can be rotated around the center in any direction. The rotation change is displayed in the lower left part of the 3D view window. The campaign must end with [LMB] or [enter] or aborted with [RMB] or [esc].

### Scale

The scale action is started with the [s] key. Use the mouse to take control of the object and it can be uniformly enlarged or reduced. The resize is displayed in the lower left of the 3D viewport. The campaign must end with [LMB] or [enter] or aborted with [RMB] or [esc].

### Limiting transformations

All transformations have in common that they can be limited to operating along a selected global or local coordinate axis. This is achieved by pressing the [x], [y] or [z] keys after activating the transformation to limit along the global coordinate axes or press the same keys twice to constrain along the local coordinate axes. The operation of transformations can also be limited to selected spatial planes. This is achieved by pressing [shift] + [x], [shift] + [y] or [shift] + [z].

### Reseting the transformations

Transformations can be reset to default values by holding the [alt] key next to the individual transformation key, ie: [alt] + [g] resets the position, [alt] + [r] resets the rotations and [alt] + [s] resets the scaling.





#### Precise transformations

Each transformation can also be precisely quantified. This is done by entering the numeric value for which you want to perform the transformation after activating the transformation.

E.g., press [g] then [x] and then enter the value of the displacement along the x axis and confirm the transformation or e.g. press [r] enter the value for x, press [tab] and enter the value for y and press [tab] again and enter the value for z and confirm the transformation.

Two tools can be used to precisely determine the transformations using the mouse: holding down the [shift] key slows down the execution of the transformation, and holding down the [ctrl] key performs the transformation in discrete steps. However, holding both keys down reduces the discrete step.

### Transformation manipulators

For approximate transformations of objects in the scene, we can also use t. i. transformation manipulators shown in the figure below.



The first manipulator is designed to move objects, which is achieved by grabbing an individual arrow / tile with [LMB] and moving the object along the selected axis. The second manipulator is designed to rotate the object, which is achieved by grabbing an individual circle with the mouse and rotating it. The third manipulator is designed to scale objects and works

the same as the first. The last manipulator combines all three modes simultaneously. You can switch between manipulators with the buttons at

the bottom of the 3D view window. You can activate several modes at the same time by holding down the [shift] key while clicking on individual buttons. Individual manipulators can also be selected from the quick menu accessible by pressing [ctrl] + [space].

### Transformations using UI

The first manipulator is designed to move objects, which is achieved by grabbing an individual arrow / tile with [LMB] and moving the object along the selected axis. The second manipulator is designed to rotate the object, which is achieved by grabbing an individual circle with the mouse and rotating it. The third manipulator is designed to scale objects and works the same as the first. The last manipulator combines all three modes simultaneously. You can switch between manipulators with the buttons at the bottom of the 3D view window. You can activate several modes at the same time by holding down the [shift] key while clicking on individual buttons. Individual manipulators can also be selected from the quick menu accessible by pressing [ctrl] + [space].



1 1

🦯 Global





### Object aligned view

The view in the scene can be aligned with the active highlighted object by using the [shift] key (e.g., front view [shift] + [num 1]) when switching to aligned views.

The view can be focused on the active object also in the orientation in which it is. This is achieved with the [num,] key.

We can also lock the view of the active object when we move it around the scene. This is achieved with the shortcut [shift] + [num,], the view is "unlocked" with the key combination [alt] + [num,].

. . .





# Lecture 02 Introduction to 3D modeling

### Overview

- 1. What is 3D modeling?
- 2. What 3D modeling techniques are there?
- 3. How do we learn 3D modeling?
- 4. Basic shapes
- 5. Modeling with primitives

# What is 3D modeling?

3D modeling is the process of making a 3D representation of an object in a virtual world. Such objects can then be used in many ways: for 3D printing, for use in images, videos, games, as plans for manufacturing in industry, and more.

The closest equivalent of 3D modeling in the real world is sculpture, which differs from 3D modeling due to several factors: in 3D modeling it is not necessary to choose the material from the beginning (if at all), in the virtual world we are not limited by real world limitations. where we cannot make objects that do not satisfy physical laws or self-intersecting models, in the real world we are limited by the set of tools we can use to create as well as the shapes we can achieve with such tools.

It is believed that everything we can create with 3D modeling cannot be recreated in the real world. It is also true that 3D models created in the virtual world can be easily animated, which is especially important for video and interactive content.

## What 3D modeling techniques are there?

There are many different 3D modeling techniques and many ways that define the workflow. During the lecture, we will learn about two basic 3D modeling techniques that are suitable for beginners:

- 1. 3D modeling using curves.
- 2. 3D modeling using polygons.

In addition to the above, there are many other techniques that are also supported in Blender (e.g., virtual sculpture, meta-sphere modeling, surface modeling), as well as techniques that Blender does not support and are intended for specific applications (e.g., full-scale modeling). geometry). Different modeling techniques provide different tools. While in Blender we can prepare a model for





3D printing, we cannot prepare models in the form expected by an industrial plant (e.g., in the automotive industry), because the requirements are simply different.

However, we can often switch between different modeling techniques during the process itself and thus use the most suitable one to achieve the desired results in a certain step.

The techniques we will learn about in the workshop are very useful for modeling general scenes, as well as e.g. for modeling objects used in the computer game industry, in the production of video content as well as for the idea of final representations.

# How do we learn 3D modeling?

Practice, practice and practice again.

As with any other thing, it is also true for 3D modeling that it is not enough just to know the tools and techniques, but in order to successfully achieve the desired shape, we also need to practice a lot, i.e., a lot of 3D modeling. When we master a 3D modeling technique well enough, we can achieve almost any results with it. However, by mastering several different techniques, we can achieve the desired results faster and more efficiently.

### Basic shapes

Pure bases include the creation of 3D models from the basic primitives at our disposal (e.g., surface, cube, circle, sphere, cylinder, cone, torus and various curves).

New objects can be added via the Add menu.



In the following, we briefly present the individual basic primitives:

• the plane represents a level limited part of a plane, which basically consists of four vertices and one quadrilateral;





- the cube represents the mathematical shape of the cube, consisting of 8 vertices and 6 faces;
- a circle represents a primitive of a circle defined by a finite number of connected straight lines;
- sphere (UV sphere and ICO sphere) represents a primitive sphere composed of quadrilaterals in two different ways or with vertical / horizontal division in the case of a UV sphere or in the form of an ICO primitive of the desired degree;
- the cylinder represents a grid representation of the shape of the cylinder, to which we can determine the number of vertical divisions and what kind of cover (if any) it has;
- the cone represents a grid representation of the shape of the cone, to which we can determine the number of vertical divisions, what are the radii above and below (the upper is usually 0) and what lids are at the ends of the cone;
- the torus represents the grid geometry of the torus (donut with a hole), to which we can determine the number of divisions along both dimensions and the main and transverse radii;
- curves represent a multitude of different curves, which we will get to know in more detail later;
- the monkey represents a model of the monkey's head, which is the trademark of the Blender program.

We can change the basic properties of primitives until we start editing them with transformations or even changing grid geometry. To get to the basic settings, press the [f9] key after creating the object when it is highlighted. A pop-up window opens where different basic parameters are available for different primitives. In the image below we can see the set of parameters available to us for the UV sphere.

# Modeling with primitives

There is a lot that can be done with primitives. For the exercise, try making a snowman as shown in the picture below.







The primitives used to make the snowman are: (ICO sphere, cylinder, UV sphere and cone). A little more work is needed to assemble a simple model of the castle from the primitives shown in the figure below:



In this case, a different set of primitives is used (plane, square, cylinder and cone), but of course there are significantly more of them than in the previous case.





# Lecture 03 Modeling using curves

### Overview

- 1. What are curves?
- 2. Why curves?
- 3. Draving Bezier curves
- 4. Creating geometry using curves

### What are curves?

Curves are mathematically defined uneven continuous lines in space. They are more precisely defined by parametric functions, which define the position of an individual point on a line depending on an input parameter (usually named t). There are several classes of curves that differ from each other mainly in how we define them.

The curves were invented by French engineers de Casteljau and Bezier (the former at Citroen, the latter at Renault) for the purpose of accurately defining the shape of a car body in the 1960s. Even today, Berzier curves (more precisely cubic Bezier curves) are among the most used curves in design. They have been encountered by virtually anyone who has ever opened a vector design program (e.g., InkScape or Adobe Illustrator), as well as many others.



# Why curves?

Curves allow us to define rounded smooth shapes, which are often used in the design of organic shapes. Curves are also useful for defining smooth paths along which objects can be guided during animation. Curves can also serve as a good basis for constructing geometries from pre-prepared sketches.

In the following, we will look at how bezier curves can be used to model 3D shapes.





## Drawing Bezier curves

Drawing and editing curves in Blender is a bit more complicated than in dedicated 2D design programs. This is partly due to an additional third dimension, and partly due to a different philosophy of interaction.

In Blender, in addition to the possibility of creating the usual network geometry, we also have the possibility of creating Bezir and NURBS curves and circles. In the workshop, we will only learn about the use of Bezier curves and circles.

Add Object		
▼ Mesh		
⊃ Curve	٣	°℃ <u>B</u> ezier
Surface		ိတ် <u>C</u> ircle
🐨 Metaball		TC Nurbs Curve
a Text		O Nurbs Circle
🔔 Volume		∡ Path
ດ Grease Pencil		
* Armature		
# Lattice		
Empty		
Image		
👳 Light		
1 Light Probe		
Gamera		
Comera		
()) Speaker		
Force Field		
Collection Instance		

### Curves and circles

New curve or new circle is created by clicking the Add button in the upper screen area of the 3D view and selecting the Bezier curve or circle. The same is achieved by pressing the [shift] + [a] key combination above the 3D display to trigger the add command. This opens a pop-up menu where you can also select curves from the submenu. The difference between a curve and a circle is that the curve is not closed by default, but a circle is in fact a closed curve with appropriately spaced control nodes.

When we create a curve, it appears at the location of the 3D cursor and has a default shape. We can move such a curve around the world, we perform transformations over it as over all other objects. For more detailed editing of curves, it is necessary to switch from object mode to edit mode (combination [tab]).



In the curve editing mode, other tools are available to us and we have access to the components of the curves - control nodes and levers that allow us to edit the shape of the curves. The curve normals (arrows along the curve) are also displayed by default, illustrating the orientation of the curve.

When the display of normal and curve levers is superfluous, we can turn it off in the overlays option under the *Curve Edit Mode* section, where we find the *Normals* control.

Just as transformations can be performed in object mode, transformations over individual components of curves can be performed







in edit mode. Control nodes and levers can be labeled in the same way as objects, and shortcuts to transformations that work the same way are the same.

### Deforming curves

Deformations are available to us when editing curves in menus at the top of the 3D viewport. The most commonly used tools are:

- subdivide, allows us to add additional control nodes among the existing nodes on the curve and thus a more detailed definition of the shape of the curves; the command is triggered with the [w] key and new nodes are added between the marked control nodes, while maintaining the shape;
- smooth, allows us to smooth the shape of a curve by reducing the distance between the control nodes without changing the positions of the control levers; the command is also triggered by the [w] key and reduces the distance between the selected nodes, thus smoothing the appearance of the curve;
- extrude, allows us to extend the curves from the final control nodes. This creates a new or. new nodes at one or both ends of the curve; the command is triggered with the [e] key and then with [LMB] we determine the position where we want the new node or set of nodes;
- handle type changes the type of control levers for selected nodes; the command is triggered with the [v] key and allows us to choose between the types: Automatic, Vector, Aligned and Free; different types of levers allow us to control the course of the curves in the control nodes differently (whether the curves will be smooth or broken);
- toggle cyclic, allows us to change an unclosed curve to a closed one and vice versa; the command is also triggered by pressing [alt] + [c];
- make segment, allows us to connect two unrelated parts of the curve; the command is triggered with the [f] key and a new part of the curve is created between the two finite end nodes;
- separate, allows us to divide the selected curve into two objects, which is achieved by marking the points of the curve that we want to separate into our own object and trigger the command with the [p] key; this creates a new object that contains a curve defined by the selected points;
- delete, allows us to delete individual control nodes or parts of the curve; trigger the command with the [part] or [x] key, then select what you want to delete; in the case of deleting carbon, the curve will remain connected but will change shape accordingly, and in the case of deleting segments, we will delete part of the curve and thus break it on the selected part.





With the presented tools, the curves can be easily transformed into the desired shape. Of course, we are not limited to two dimensions, but we can also create curves in three dimensions. Thus, we can create a rather complex 3D object from a simple curve as shown in the figure below.



# Creating geometry using

There are many ways to create geometry using curves. In the workshop, we will learn just a few basic examples of creating geometry.

### Creating using extrude and bevel

From the curves, the t can be used. i. extrude extract a flat surface from the curve. Such an example is shown in the figure below, where a surface is extracted from a closed Bezier curve. Of course, we can determine the amount of extraction as well as how accurately the curve will be defined by individual parts. All of this is available in the Property Editor within the Curve Properties tab.

The amount of extraction is determined in the *Geometry* section by changing the *Extrude* parameter, as shown in the figure above. The result is shown in the figure below. If you want to include the display of individual plots, you can also achieve this in the Property Editor in the tab with the



properties of the object, where in the *Display* section include the property *Wire*.







In addition to the extraction, we can also add a bevel to the model, to which we can determine both the depth and the resolution. An example of the use of bevel and extraction is shown in the figure below.



Instead of oblique planes, they can also be defined by a curve. In this way, we achieved the already shown geometry of the curved connected pipe again shown below. In this case, we chose a circle for the prototype oblique curve.



The surface extracted from the curve has a thickness of 0 and as such it is difficult to illustrate a real object. From flat surfaces, volume geometry can be created using modifiers. Some of the selected ones are presented below.

### Using modifiers with surfaces and curves

Modifiers are used in Blender for many things: to modify a grid, to generate things, to deform things, and to simulate. In the following, we will look at two modifiers designed to generate geometry from curves and surfaces:

- screw, allows us to rotate the selected curve around an axis and thus create a spindle; working with a bag tool is quite similar to real work with a lathe;
- solidify, allows us to thicken a selected surface into 3D geometry (e.g., square thicken into a square); the modifier is particularly useful in the case where e.g., by screwing we create an open geometry that is really just a surface.





### Modeling using curves

With curves, geometry can be constructed in many ways by combining the approaches presented above. Using the curves, try to create a cup as shown in the image below. All it takes is two Bezier curves and two Bezier circles. Make this using a prototype curve to create a curve bevel (a circle is a bevel from a curve).



Next, try making a candlestick as shown in the sketch below. Use one curve and a screw modifier to design it.







# Lecture 04 Polygon modeling

### Overview

- 1. Reference images
- 2. What are polygons?
- 3. Why polygons?
- 4. Creating polygon models
- 5. Editing polygon geometry
- 6. Examples

# Reference images

When making 3D models, we often benefit by using so called reference images. They can be used to: (1) facilitate the idea of the final product, (2) be an aid to the definition of details, or (3) as an auxiliary sketch in the modeling itself. While for the first two scenarios we can use a variety of photos and / or sketches that we have found / produced and open for help in the image viewer on our system, for auxiliary sketches / images in modeling, they should generally be aligned with the view as it is used in modeling.

In most cases, we benefit from sketches / pictures / photographs of standard views (e.g. front view, side view, rear view, top view and / or bottom view). In addition to these, the views aligned with the details on the buildings also benefit us in the production of details.

In Blender, images can only be placed in the background of the view in the orthographic camera mode (parallel projected images), aligned with the selected axis or multiple axes simultaneously. When you turn the camera out of alignment, the background image also disappears. To add an image to the background, add the image  $Add \rightarrow Image \rightarrow Background$ . When we click on the button, a selection window opens, where we find the appropriate image on our computer and confirm its selection. The image is displayed in the current view. In addition to aligned views, the background image can also be used in the view through the camera.

Similar to background images are reference images, with a difference that they are always visible, easily selectable and modifiable as regular objects in the scene.

We can change several properties of the background images:

- align them with the selected *axis*;
- we can include them in the file with 3D model (*File -> External Data -> Pack resources*);;
- we can set their transparency;
- we can put them in from or in the back of all the scene objects;
- we can move the arround;





- we can mirror them across vertical or horizontal axis;
- we can rotate them around the center;
- we can scale them.

By adjusting the presented properties, we can customize the display of auxiliary images to aid modeling. With more advanced use, videos can also be a source of background images.

# What are polygons?

Polygons or polygons are figures with three or more vertices. In fact, all polygons with more than three vertices are transformed into triangles before drawing, and only then are they drawn. The reason for this lies mainly in the design of graphic pipelines and the need for a uniform definition of objects.



Polygon geometry is still the predominant way of presenting 3D objects and is most prevalent mainly due

to its use in 3D interactive computer graphics (e.g., in computer games). As a result, many

different tools and gadgets are available with the design of such 3D geometry. In the following, we will look at how such geometry can be formed at the lower level - by moving individual corners.

An individual triangle consists of basic building blocks: (1) corners, (2) edges) and (3) surfaces or. cheeks, which is also shown in the picture on the right.

As we have already learned, Blender offers us a set of basic

primitives for starting from 3D modeling (cube, cylinder, cone, ...). Most of these building blocks are defined by polygons. In modeling using polygons, we take as a starting point the closest primitive in terms of shape, which is then formed into the final desired shape.

# Why polygons?

As mentioned in the previous chapter, the main reason for their use is their high prevalence and consequent adaptation of both software and hardware. Last but not least, the graphic processing units are highly optimized just for drawing 3D geometry presented with polygons.

A very positive feature is the compactness of the presentation. More complex 3D geometry can be designed from polygons by using a larger number of polygons in places where we want more detail, but where there is no such need, we can use a smaller number of polygons.

They are also very convenient due to the fact that it is very easy to "glue" textures to polygons images that represent a more detailed structure of objects and also determine their color and appearance. Over the years, many techniques have also developed that allow us to add additional virtual details to otherwise roughly defined 3D models (e.g., using bumps, offsets, normal

22 | Ciril Bohak @ FRI | 3D modeling using Blender







mapping). There are many more advantages, but we will get to know only a small selection of them in the workshop.

# Creating polygon models

To create polygonal geometry, we need to start with one of the primitives available to us. The simplest is the primitive of the plane, represented by four vertices, 4 edges, and one surface. In order to create new polygons, we need to change such initial geometry in the editing mode, which is presented in the next chapter.

As a starting primitive, it makes the most sense to choose the primitive that best represents the shape and properties of the desired finished product. To make a face, we can start with a cube or a sphere (both are a good starting point), for elongated objects it is usually best to choose a cylinder or an elongated square, etc.

## Editing polygon geometry

To edit the polygon geometry, select the 3D object and switch to edit mode (shortcut [tab] + [1,2,3]). In edit mode, we have the ability to edit corners, edges and / or faces. We can

edit only individual components or combine them by editing different components. Control over individual components is switched on and off with the buttons in the 3D view menu below. Editing several components at the same time is achieved by using the [shift] key when clicking on the buttons.

### Editing vertices

Vertices are the most basic components that represent 3D geometry. If our basic geometry consists of enough vertices and they are properly interconnected, we can create the desired final shape just by moving the vertices from a primitive. In most cases, of course, this is not the case, so modeling tools are used to help us add new vertices or merge existing vertices in various ways. We will look at some of these tools below. All editing tools are available in the *Vertices* menu of the 3D view window in edit mode. The same menu can also be called up with the shortcut [ctrl] + [v]. Some basic tools are:

• Merge - merges marked nodes to the selected location (first position, second position, center or cursor position). If we mark all the corners of a face, that face will disappear.



Vertex				
Extrude Vertices				
Extrude to Cursor or Add Ctrl Bevel Vertices	Right Mouse Shift Ctrl B			
New Edge/Face from Vertices Connect Vertex Path Connect Vertex Pairs				
Rip Vertices Rip Vertices and Fill Rip Vertices and Extend				
Slide Vertices Smooth Vertices Smooth Vertices (Laplacian)				
Vertex Crease				
Blend <u>f</u> rom Shape Propagate to Shapes				
Vertex Groups Hooks	Ctrl G► Ctrl H►			
Make Vertex Parent				







- Remove Doubles removes duplicates from selected corners. You can also set the maximum distance for detecting duplicates.
- Extend Vertices the tool allows us to add a new face to a specific face by pulling a new node from the highlight vertices.
- Connect Vertex (Path) is a tool designed to connect unconnected corners. Mark the



vertices you want to connect with the edges and use the shortcut [j].

- Slide allows you to move the selected slide along the connected edges of the grid.
- Bevel allows you to add new corners from one corner along the connected edges to create a bevel or just divide the geometry.



• Smooth Vertex - Reduces the angles that surround the node with the connected edges.



### Editing edges

Similar to the basic set for working with corners, we will also present the basic set of tools for working with edges. Only by using basic transformations (displacement, rotation and scaling) can the existing geometry be transformed into the desired final shape, if we have enough and correctly connected edges. They can be accessed via the *Edges menu*. with the shortcut [ctrl] + [e]. The basic tools that help us add and / or subtract / combine edges are:

- Create Edge allows us to connect marked unconnected edges to each other and create missing edges and faces.
- Subdivide allows us to divide the selected edge evenly



into the desired number of shorter edges.

• Bevel - divides the selected edges with a slash.



• Edge Slide - allows you to move the selected edge along the connected edges.

### Editing faces

Just like editing edges, you can also use basic transformations (shift, rotation and scaling) to edit faces. In addition to them, the following tools are among the most useful via the Face menu or. with shortcut [ctrl] + [f]:

- Insert Faces adds new faces to marked slaves.
- Bevel splits the connected edges with a bevel.
- Poke Faces shares marked faces with connections between all corners.





Edge	
Extrude Edges Bevel Edges Ctrl B Bridge Edge Loops Screw	
Sub <u>d</u> ivide Subdivide Edge- <u>R</u> ing Un-Subdivide	
Rotate Edge <u>C</u> W Ro <u>t</u> ate Edge CCW	
Edge Slide Loop Cut <u>a</u> nd Slide Ctrl R Offset Edge Slide Shift Ctrl R	
Edge Crease Shift E Edge Bevel Weight	
Mark Seam Clear Seam	
Mark Sharp Clear Sharp Mark Sharp <u>f</u> rom Vertices Clear Sharp from Vertices	
Mark Freestyle Edge Clear Freestyle Edge	

Face	
Extrude Faces	E
Extrude Faces Alon	g Normals
Extrude_Individual F	aces
Inset Faces	
Poke Faces	
Triangulate Faces	Ctrl T
Tris to Quads	
Solidify Faces	
Wireframe	
Fill	
Grid Fill	
Beautify Faces	
Intersect ( <u>K</u> nife)	
Intersect (Boolean)	
Weld Edges into Fa	ces
Shade Smooth	
Sh <u>a</u> de Flat	
Face Data	





- Triangulate faces turns all marked faces into triangles. This benefits us when we want to represent the final geometry only with triangles.
- Tris to Quads Tries to turn all marked triangular faces into quadrilaterals.

For all components, you can also use the extraction tool, which is triggered by the shortcut [e].

### Smoothing the polygon geometry

Because the production of models is extremely demanding to the smallest detail, and we mostly want the final representations not to show the "rough" of the mesh geometry, in modeling we use model smoothing in most cases. For the purposes of modeling, we can use temporary smoothing and thus already during the creation we see how our network will be smoothed, and it is easier to add the desired details.

Switch between the different levels of smoothing with the shortcuts [ctrl] + [0]... [ctrl] + [9]. This adds a modifier to our geometry, which can be set separately to which level the geometry will be smoothed in the view and to which level when rendering.





# Examples

### Candlestick

Use the tools presented to try to create the candlestick below.







### LEGO figure

Make a LEGO figure based on the pictures below.



. . .





# Lecture 05 Materials

### Overview

- 6. What are materials in 3D modeling?
- 7. What are textures?
- 8. Node editor
- 9. Material types
- 10. UV mapping

# What are materials in 3D modeling?

So far, we have learned how we can create different 3D objects with different modeling techniques. In this chapter, we will look at how to add colors to such objects. The colors of objects are determined by their materials. With materials, we mostly want to illustrate the appearance of objects in reality as convincingly as possible, thus not determining the physical properties of objects.

The materials used are only used to display objects during the rendering process, as too much computing power is required to draw them correctly, and consequently this cannot be done interactively on ordinary personal computers.

The materials represent the implementation of a mathematical model for calculating the illumination of objects in the scene, which try to imitate as realistically as possible what is happening in the real world, which is also called physically based representation. The types of materials available to the user also depend on the imaging drive used, as different drives use different sets of properties when calculating.

In the following sections, we will only look at the materials supported in the *Cycles Renderer* rendering drive, which is already included with Blender by default and replaces the default *Blender Renderer* drive, which is no longer in development. In the following - in the property *editor -> Render* properties, select *Cycles Renderer* instead of *Eevee* in the Render Engine drop-down menu, and we will learn more about the rendering below.

# What are textures?

In the world of 3D computer graphics, textures are images that contain information about the surface of objects (e.g., colors, offsets, surface orientation, gloss level, fusion coefficient, etc.). They can be two-dimensional (which we will deal with the most) or three-dimensional. In order to use





the information stored in the texture in the rendering, we need to define which part of the surface corresponds to which part of the texture, which is called UV imaging. UV mapping tells how a 2D texture is mapped to the surface of a 3D object, which we will look at below.

## Node editor

Most properties related to textures and materials are edited in the *Shader Editor* in Blender. You can access it by selecting *Shader Editor* from the list of editors in the selected section of the program. This editor graphically shows us the individual components of materials (or some other properties of objects) and their interrelationships. Tools for adding new elements are equivalent to a 3D view. So we add new elements with the shortcut [shift] + [a], where a large set of widgets is available.

To add material to an object, first move to the materials tab in the Property Editor (see right image) and click, add a new location for the material (click on +) and select New. For the selected 3D object, the basic material shown in the left image is shown in the

<ul> <li>Principled BSDF</li> </ul>		<ul> <li>Material Output</li> </ul>
	BSDF 🔍	All v
GGX	<b></b>	Surface
Christonson Burlow		• Volume
Christensen-Buney	<u> </u>	<ul> <li>Displacement</li> </ul>
Base Color	_	
<ul> <li>Subsurface</li> </ul>	0.000	
Subsurface Radius	~	
<ul> <li>Subsurface Color</li> </ul>		
Metallic	0.000	
<ul> <li>Specular</li> </ul>		
<ul> <li>Specular Tint</li> </ul>		
Roughness		
Anisotropic		
<ul> <li>Anisotropic Rotation</li> </ul>	0.000	
<ul> <li>Sheen</li> </ul>		
<ul> <li>Sheen Tint</li> </ul>		
Clearcoat		
Clearcoat Roughness		
• IOR		
<ul> <li>Transmission</li> </ul>		
Transmission Roughness		
e Emission		
<ul> <li>Alpha</li> </ul>	1.000	
Normal		
<ul> <li>Clearcoat Normal</li> </ul>		
<ul> <li>Tangent</li> </ul>		

Node Editor. As a rule, we can connect the outputs and inputs of individual elements of the same color (exceptionally also differently). Below we will take a look at some of the basic materials we can create in Blender.



# Material types

In general, Blender has an almost unlimited number of materials that can be created from basic models for calculating lighting. Basic lighting models are also called shaders. The main groups of sunshades are:

- Surface Shaders designed for objects where we are interested in how light is reflected from the surface. This includes most common materials;
- Volume Shaders intended for objects through which light passes (transparent and translucent objects). These include materials where light penetrates the surface (smoke, fog, water, etc.);





• Displacement - are intended for cases where we want to deform the surface of objects using additional data (e.g., textures).

#### Surfaces

The shading calculation on the surfaces of buildings is performed in Blender using the BSDF (Bidirectional Scattering distribution function) function, which describes what happens to the rays on the surface of the material:

- the rays can be completely reflected and so we get a mirror effect;
- the rays can be scattered, which in most cases gives the object color;
- the rays can be refracted through the object, which is the case with glass, for example;
- the rays can be reflected and scattered several times below the surface of the object, which conjures up transparent and translucent objects (this includes, for example, human skin).

In the following, we will present some typical materials for everyday use in 3D modeling.

#### Metal materials

The main feature of metallic materials is their luster. For such purposes we use Glossy shader, which reflects most of the light, but we can determine how rough it is on the surface and what proportion of light is scattered outside the perfect reflection. If we set the surface to be completely



smooth, we get a mirror. An example of the composition of the nodes of such material is shown on the right, and in the illustration it is located in the middle of the figure at the end of the section.

#### Plastic materials

Plastic materials (even more precisely rubber) scatter most of the light and so we do not see reflections of neighboring objects on them. For such purposes we use Diffuse shader. The surface of such materials is illuminated very evenly no matter what



angle the light falls on it. An example of such a material is created each time we create a new material and it is shown to the right, and in the final representation it is to the right of the metal (4th object in 2nd row).

#### Glass

It is characteristic of glass that light travels through the material and the path of light is refracted. The fracture depends on the fracture coefficient of the material. For glass we use Glass shader, which allows us to adjust the refractive index and is shown on the right. In the final representation, it is shown to the left of the middle object (2nd object in the 2nd row).





#### Transparent and translucent materials

Transparent (and translucent) materials transmit light but do not necessarily refract it. In addition, light in such buildings is also scattered inside.

Transarent, Translucent and Refraction sunshades



▼ Transparent BSDF	<ul> <li>Material Output</li> </ul>
BSDF	All 🗸
• Color	Surface
	Volume
	<ul> <li>Displacement</li> </ul>

are used to define such materials. An example of a translucent material that does not refract light is shown on the right. In the final representation, the translucent material is used on the 5th object in the 2nd row and the refraction on the 2nd object in the upper row.

#### Combined materials

For a material that combines the properties of different basic materials, individual materials can also be combined using Shader mix. It accepts two sunshades at the output and merges them into the final output according to the given weight. An example of the use of such a sunshade is shown on the right, and in the final illustration, combinations of different sunshades are used on the 3rd object above (combines Diffuse and Glossy), 1st object in the middle (combines Diffuse and Refraction), 4th object in the bottom row (combines Galss and Glossy) and at the last facility below (combining Diffuse and Ambient Occlusion). Materials can also be combined in several levels, thus bringing the properties of any number of basic materials into the final material.

A number of other basic BSDF sunshades are available to users, but they are more specific. One such is e.g., Toon shader used to achieve a cartoonish depiction.



### Volumes

The calculation of shading in volumetric objects is even more demanding than the calculation on the surfaces of objects. Volumetric objects are, for example, smoke, clouds and fog. What they have in common is that light passes through such objects and is reflected from small particles inside

Voronoi Texture	▼ Volume Scatter		
Distance 🔶	Volume		
Color 🔶	- Color	Surface	
Position 🔶	Density	Volume	
3D 🗸	Anisotropy 0.000	Displacement	
F1 ~			
Euclidean 🗸			
Vector			
Scale 5.000			
Randomnes 1.000			





the objects or is refracted on them. In order to depict correctly, it is necessary to simulate the passage of a large number of rays through such objects so that we can correctly calculate their color and transparency. A simple example of a balloon made from a slightly reworked sphere with the material defined according to the right image is shown below.



### Displacements

The use of displacements is intended to add additional details to existing building surfaces that we do not want to add with modeling techniques, or we can even generate them automatically and save time, while allowing the possibility to correct such details on the surface later. Details can be added to the surface of the building in three ways:

- using the displacements technique on geometry, where details can be added directly to the geometry of objects, which, however, requires that the grid of the model is as dense as we want the details to be accurately presented on it;
- using the displacements technique on surfaces, where the details can be added directly to the surface of the material without the need to adjust or change only the geometry of the object;
- using normal maps that contain information about the detailed orientation of the surface of objects (Normal maps), and this information is used only in the rendering.

### Using the displacement technique on geometry

For the displacement technique, it is necessary to further divide the grid of the building as densely as we want the precise details to be added. This is accomplished by using the Subdivision Surface modifier found in the modifiers tab in the property manager. For example, we will add bulges in the form of waves to the plane, as shown in the example below.









We first added a division modifier to the plane, changed the type of division to Simple, which does not smooth the grid. Then we added a Displace modifier. For use, we added a new texture of the Clouds type, where we chose Perlin's noise as the base and played around with the settings a bit. For the final effect, we also reduced the impact strength of the offset modifier.

In this way, you can also define what additional details on the surfaces of buildings should look like with any texture.

#### Using the displacements technique on surfaces

With the technique of deviations above the surfaces, we achieve the addition of small details to the surface of buildings where we do not want to add additional network geometries. In this way, bulges and irregularities can be introduced into the surfaces at the lowest level. To add such details, it is necessary to adjust the material of the object as shown in the figure to the right. The result of such deviations added to the deviations above the geometry is shown in the figure below.





#### Using normal maps

Normal textures are used in most cases to add details that are reflected in how light is reflected on a particular part of the surface. In this way, in addition to deviations, we also change the shading calculation of buildings and in this way we achieve fine details such as dents and scratches on the surfaces of buildings or the detailed texture of the material.

As an example, we added details defined by the Voronoi procedural texture to the waves we defined in the previous example. To do this, we had to adjust the material of the building as shown in the figure below.





Two results of the use of normal textures are shown in the figures below: the right for the above example and the left for the case where the Voroni pattern is used for both normal texture and offsets.





# UV mapping

The basic purpose of UV imaging is to map the applied texture to the surface of the object. UV is derived from the naming of coordinate axes in the texture space (it is two-dimensional). To be able to do this it is first necessary to perform UV unwinding, which is a process in which the network of the building is unwound from the building itself and mapped to a 2D surface.

Blender offers us quite a few pre-prepared methods for UV unwinding. All methods related to UV unwinding are available via the shortcut [u] or. via  $UV \rightarrow Unwrap$  menus. Additional tools for marking cuts above the grid are also available between tools above the edges [ctrl] + [e] or. via the *Edges* menu. Among the edge tools, the *Mark Seam* and *Clear Seam* commands are particularly suitable for unwinding purposes. These tools are available to us in edit mode.

For unwinding purposes, it makes sense to open the *UV Editor*, where you can see the texture that will be stretched to the selected geometry as well as which part of the texture will be mapped to which face of the grid.

Within this lecture, we will present two ways:





- UV unwrap try to spread the mesh geometry along the given seams and project it on the plane. If we did not add seams to the grid then all the cheeks spread over the entire texture;
- 2. smart UV projections tries to unwrap the model grid automatically and project it onto the plane.

The standard course is as follows:

- 1. we first open the UV Editor and create a new texture in it (preferably a UV Grid or some texture of our own);
- 2. we adjust the texture to the material of the object;
- 3. switch to texture display mode;
- 4. switch to object editing mode;
- 5. we define the seams of the geometry;
- 6. use one of the unwarp methods presented above;
- 7. individual components of the developed geometry can also be moved in the UV Editor until the desired layout is achieved.

More advanced UV unwarping is a rather complex process that requires a lot of manual work. It is only when we have the geometry properly unwound that it makes sense to start preparing the appropriate textures.



. . .





# Lecture 06 Lights

# Overview

- 1. What are lights and what types of lights are there?
- 2. How to place the lights in the scene?
- 3. Objects as light sources

# What are lights and what types of lights are there?

The lights in Blender are objects that represent light sources as we know them in the real world. The main types of lights we have available in Blender are:

- Point light source represents a light source at one point in space from where it propagates evenly in all directions. We can define its strength, color and how the strength decreases with distance, as well as a number of other parameters;
- Directed light (Sun Blender) is a ubiquitous light source that has no defined source but has a defined direction. In addition, we can also set the strength and type of other parameters. It is mostly used to simulate sunlight and consequently we can also adjust the properties of the atmosphere and the sky;
- Spot light light originates at a certain point and spreads into space in the form of a cone. We can define what shape the cone is and where it is directed, as well as how sharp the transition between full illumination and darkness is;



• Area light - represents a source of light emanating from a specific area. A good example of such an origin is the computer screen. In the Blender, they can also be used as light sources that penetrate the room through windows, doors and other openings.



While when using the default Blende Renderer for each type of light, you can set most parameters in the property manager, when using Cycles Renderer, this is not the case, as we have to work with nodes that we have learned when using materials to set the intensity of each light. The light settings can be accessed by

of each light. . The light settings can be accessed by selecting the third tab from the left in the property editor. The type of light can be selected during the addition, but it can also be changed later in the properties. Different lights can be accompanied by different properties such as size, whether the light casts shadows or not, what shape the light is, etc.



blender

For a more detailed understanding of the operation of lighting models, I suggest you read the article entitled Lighting models published in the journal  $\check{Zivljenje}$  in tehnika (5-2017).

# How to place the lights in the scene?

For the most realistic lighting, it is necessary to place the light sources in those places where the light would actually originate. We try to use as similar a source as possible, which is available in Blender, for individual sources. Sunlight can be modeled with a directed type of light, bulbs with point and reflector sources, larger lights (eg neon lights) with field light sources. Care must be taken to adjust the light intensity accordingly.

If we are planning an animation, it is necessary to think properly about moving the source and changing its intensity. By using nodes, we can also create quite complex light sources that throw patterns into the room, different colors depending on the angle, etc.

### Simple example setup

For cases when we want to present a 3D model well, in practice the standard scene lighting, which includes three point light sources:

- one source in front to the left of the object we want to illuminate;
- the second source is in front to the right of the object;
- move one of the light sources slightly above the middle height of the object, the other slightly below the middle height of the object;
- with these sources we achieve good illumination of the object from the front;



• for added effect we can experiment a bit with the intensities and colors of the lights;





• the third light source, which is typically much weaker, is placed behind the object, thus emphasizing the outlines of the object.

### How to additionally tune the lighting?

 Change the size of the shadows - this is determined by the size of the light source: the larger the source, the softer the shadows will be, the smaller the source, the sharper they will be. Soft shadows bring serenity to the scene, while sharp shadows accentuate the details on the objects. In the example below, the comparison between the light source size is 10.0 (left) and 0.1 (right);



2. Don't overdo it with the number of lights - using too many light sources eliminates shadows, which play an important role in recognizing details. Without shadows, the geometry looks perfectly flat. Below is a comparison between using one light (left) and many lights arranged in front of an object (right);



3. Use colored lights - with colors you add an extra emotional impact to the scene, add warmth or a feeling of coolness. Real light sources are defined by the heat of the source (in Kelvins), which can be simulated using a black body when creating a shader. The example below shows the difference between using an all-white light (left) and an orange light (right);



4. Emphasize the object of interest - highlight the object you want to emphasize to attract the viewer's focus. In the example below, use a point light source (left) or a reflector source



#### (right);





5. Add textures to the lights - this breaks up the monotonous color of the lights and adds extra liveliness to the scene. The example below shows the use of texture on a reflector light source that further brightens the scene;



6. Animate lights - to find the most suitable lighting, you can animate both the positions and other parameters of the lights and thus find the combination that best serves your purpose.

## Objects as light sources

In addition to lights, the source of light can also be objects themselves. most often these are phosphorescent objects. Radiation of objects is achieved by adapting the appropriate material to them, where we choose emission shader. In this way we can create interesting effects as shown in the image below.







# Lecture 07 Rendering

### Overview

- 5. What is rendering?
- 6. Environmental maps
- 7. Rendering parameters

# What is rendering?

Rendering is the process of calculating the illumination and shading of a scene according to the properties of objects and their materials. There are many methods for calculating the exposure, but recently physically based methods have been used, which are also available in the Blender program. Blender already comes with three different renderers:

- 1. Eevee is a built-in renderer, which is quite updated from the previous version, but still has quite a few shortcomings compared to Cycles Renderer;
- 2. Workbench is mostly intended for use during modeling and scene construction;
- 3. Cycles Renderer is quite powerful and allows very realistic rendering of scenes.

Blender also supports the use of external renderers such as Autodesk Arnold, VRay, Octane Renderer and some others. The advantage of Cycles Renderer is that it is already included with the Blender program itself and is available to users completely free of charge.

## Environmental maps

When we want to place our objects in a real environment, in most cases it is necessary to model such an environment (or to use copositing). The lighting of buildings also depends on the environment, as there can be many light sources in the environment, which we did not necessarily capture by placing lights in our scene.

In such cases, we can use environmental maps, which are presented as spherical photographs captured by the high dynamic range technique. (High Dynamic Range - HDR). Such photographs also contain information about the intensity of light coming from a particular part of the environment, which can be used for the purpose of illuminating our scene. An example of a spherical photograph for 3 different exposures is shown below.







### Using environmental textures in Blender

On the World tab of the Property Editor, in the Material Properties section, click the Use Nodes button. Some new input fields appear, where, among other things, we also find the Base Color

property. By clicking on the circle next to the color, you can choose how you want to define the color. In our case, we choose Environment Texture. A set of new properties is displayed, where you can also select a file from disk (Open). We can set different parameters of the selected texture and methods of mapping to the environment. We can also determine the power of the illumination thus obtained. An example of a depiction using environmental texture on highly reflective material is shown below.

ر کړ	Q ✓ Mater	ial		0 🗅 ×	∿~
• •≺	<ul> <li>Preview</li> </ul>				
	Surface				
•	<b>-</b>		Use Nodes		
6		Surface	Principled BSDF		
♡			GGX		
•			Christensen-Burley		
	•	Base Color	• Environment Texture		
888	<b>⊾</b> • +				
			Linear		
			(=		

ĉ

5

18 18



### Rendering parameters

When rendering, it is necessary to pay attention to quite a few parameters in order to achieve the desired final appearance. In the following, we will highlight just a few of the most important ones that need special attention. Rendering parameters are available in the Render and Output Properties tab of the Property Manager, which is accessible via the camera icon.

### Renderer settings

As already mentioned, we will only get to know the Cycles illustrator in the workshop. It is a version of the renderer that uses physically based methods to calculate exposure based on Path





Tracing. Since this is an approximation method, the final appearance depends on the accuracy of the approximation, which, however, takes more time for better results.

The main parameters are:

- Device selection the user chooses whether the final rendering should be calculated on the central processing unit or on the graphics card. Depending on which piece of hardware is more powerful, we choose the appropriate option (the easiest way to check this is to perform a simple test rendering and see how long it takes on which unit);
- Resolution tells us how many image elements the final image will have. We can choose one of the predefined dimensions, but we can also determine them ourselves. The output resolution, of course, depends on the purpose of the final rendering. If the final rendering is to be displayed on an HD screen, select HD 1080, and the like for other purposes;
- File Format here we choose in what format we want the final result of the rendering. By default, this is the PNG image format, which is quite sufficient for most still images;
- Render is a key group of parameters that determines the quality of the final rendering. You can select or add a setting template (e.g., Final / Preview). At this point, only the number of samples used to calculate the illumination of a single image element is most important. The more accurate these samples are, the final approximation of the calculation method will be and the better the final product will be. The number of samples must also be adjusted to the materials used - more complex materials need a larger number of samples for better quality. The time of calculating the final rendering also depends on the set number of samples. It is recommended to use a lower number of samples for preview (16 - 64) and a larger number for the final rendering (up to 2048 samples per pixel).

Other rendering parameters are less important and you will get to know them as you get to know Blender better.

### Camera settings

Camera settings are available if you select the camera object in the scene and then switch to the camera tab in the Property Editor. The main features of the camera are:

- Lens settings where you can choose between an orthographic, perspective or panoramic version. The most common is perspective, for which we can determine the focal length (eg 35 mm);
- Camera settings where we can determine the size of the sensor used in the calculation of the image (eg 32 mm equivalent);
- Depth of Field Determines how the camera's sharpness changes with distance from the focus. In this way, we can blur objects that are out of focus, as is usually the case for





human vision and classic cameras. An example of the use of depth of field is also shown in the figure below (the blurred background is the most visible).



. . .





# Lecture 08 Modifiers

### Overview

- 1. What are modifiers?
- 2. Deformations
- 3. Generation
- 4. Simulations

# What are modifiers?

Already with the learned techniques of 3D modeling we can create any shapes and things. Many times, however, we want to make things easier and more timely. Then we resort to modifiers wherever possible. Modifiers are operations that the user can perform on 3D objects, with the ability to later correct and adjust parameters. In the following, we will learn about some modifiers intended for deformations, generation and simulation. We have already learned some basic modifiers during modeling (Subdivision surface, Screw, Solidify and Displace).

έY	Add Modifier 🗸 🗸
<b>C</b> ii	
5	
ø	
18	
۲	
$\square$	
s	
12	
۲	
6	
•	
88	

Modify		Deform	Physics
👽 Data Transfer	OD <u>A</u> rray	ズ Armature	岱 Cloth
道 Mesh Cache	Bevel	🕞 Cast	ំ <sup>ក្ត្ត</sup> Collision
🛄 Mesh <u>S</u> equence Cache	민 Boolean	) Curve	꼊 Dynamic Paint
Normal Edit	🛱 Build	🕞 Displace	🗗 Explode
_ └ Yeighted Normal	🗹 Decimate	్ర <u>H</u> ook	🖒 <u>F</u> luid
🔛 UV Project	I Edge Split	🛄 Laplacian Deform	쓙 Ocean
😸 UV Warp	() Mas <u>k</u>	🖽 Lattice	Ҡ Particle_Instance
🖙 Vertex Weight Edit	∑3 Mirror	៉ាញ់ Mesh Deform	🎲 Particle System
⊟ Vertex Weight Mix	Hultiresolution	വ് Shrinkwrap	🕼 Soft Body
Vertex Weight Proximity	(₽ Remesh	🖉 Simple Deform	
	ର୍ଥ୍ର Screw	🕝 Smooth	
	ළු Skin	Corrective	
	🗍 Solidify	ሯ Smooth Laplacian	
	O Subdivision Surface	🛄 Surface Deform	
	I <u>T</u> riangulate	🗋 Warp	
	🖉 Weld	(c. Wave	
	🗐 Wireframe		

# Deformations

Deformation modifiers deform the selected object by some predefined procedure. Some of the most commonly used basic strain modifiers are presented below.





### Simple deformations

Simple deform modifier offers users twist, bend, taper, and stretch. Each of the operations can be performed along any selected axis, which is determined by the rotation of the selected auxiliary object (usually an empty object).

The results of the individual operations are presented in the figure below, where the original is on the far left, followed by bending by  $360^{\circ}$ , then bending by  $180^{\circ}$ , narrowing by a factor of -1.5 and right stretching by the same factor.



In this way, we can quickly and easily achieve the shape we want. The direction of deformations can be defined by selecting an object in the axis, origin field that determines their orientation. In most cases, we choose an empty object that can be nested under the object that we want to deform (parent) using the shortcut [ctrl] + [p].

By transforming the deformation control tool, we can achieve an altered deformation effect and also animate the deformation.

### Lattice

Lattice is a modifier that allows objects to be modified using a three-dimensional regular grid, where each node of the grid affects the geometry in its surroundings. The user can determine how many divisions such a network has and how it adapts to the object. The figure below shows an envelope of varying degrees above the sphere object.



To create a lattice, it is necessary to create a new object of the lattice type, and then add a modifier of the same name to the object, to which we assign a control object. This allows us to use the same envelope to modify multiple objects at the same time, as well as that the modifications themselves are independent of the object and can be changed, for example, during animation. An example of different deformations by moving the control points of the same envelope over the same object is shown in the figure below.







The lattice can be set to the degree of division of the regular grid along the individual coordinate axis as well as the interpolation method by which the grid influences the change of geometry. It is basically interpolation with B-splines, but it can also be replaced by linear interpolation, Catmull-Rom interpolation or cardinal interpolation. Examples of different interpolations for the same things are shown below, where BSpline interpolation is completely on the left, followed by linear, then Catmull-Rom, and finally cardinal interpolation on the envelope.



### Warp

Warp modifier allows us to bend a part of the grid geometry towards the desired location - e.g., points in space. In this way we can visualize bulges, depressions, bumps, etc. For each curvature, it is necessary to determine the point of origin and the target point of curvature. Empty objects can be used for these two points. The modifier can also be set for how the points should affect the curvature of the surface. In addition to smooth curvature, we also have constant, spherical square, linear and others. In the figure below, the indentation of the recess is created with the help of the presented modifier, where we use empty objects for the start and end point and use a smooth type of curvature.







#### Wave

Wave modifier is designed to bend the surface using waves on the surface of objects. We can choose the axes along which the wave travels as well as the position of the wave source, which can be defined by a location in space (e.g., an empty object). The speed, amplitude, width and dimension of the waves can be determined. We can also set the damping of the waves, the time from the beginning of the wave, whether the waves are repeated, etc. The figure below shows an example of waves on a plane surface.



## Generation

Generator modifiers are designed to create new objects or. geometry based on the given input and parameters.

### Array

The Array Modifier is designed to multiply objects with a specific transformation. We can set how many times we want to multiply the selected object, determine the constant or relative deviation. The transformation between objects can also be determined by selecting another object (e.g., an empty object). In this way, we can multiply a multitude of objects from an individual object in our scene as shown in the figure below.







#### Bevel

We already used bevel in modeling. But it is also available in the form of a generative modifier. With it, bevels can be added to all edges of the selected object. Slopes can be added progressively (multiple bevels), we can determine the deviation from the base edge and the profile of the bevels. The example on the cube is shown in the figure below.



### Boolean

The modifier for combining objects is intended for combining geometries into the desired final whole. In doing so, we can choose different ways of combining: union, difference and intersection. The figure below shows all the operations above the base cubes, which are completely to the left.



### Mirror

The mirroring modifier allows us to mirror the selected geometry over the selected plane. After mirroring, the mirrored object can be combined with the original. We can also mirror over the





selected object. An example of mirroring half of Suzanne's head where the net is offset from the starting point of the object is shown in the figure below.



# Simulation

Simulation modifiers are designed to create geometry using physical simulation. We will present just a few basic modifiers that are most commonly used when creating scenes.

### Cloth

Cloth simulation allows us to recreate the physically correct behavior of clothing and other fabric products.



### Collision

The collision modifier is intended to simulate collisions between objects. Objects can move in accordance with the laws of physics (they are rigid), they can be soft, they can only represent obstacles that are taken into account when calculating collisions, etc. In this way, we can set up a sort of polygon for falling / moving objects along which we want them to move, then start the simulation and wait for the results to be calculated. Care must be taken not to perform physical simulations on too complex models and not to include too many objects in the simulation, as in





such a case the simulation will take an extremely long time. The simulation is started by pressing the Play button in the timeline and interrupted by the shortcut [esc]. The figure below shows a simulation of two soft bodies in a collision with a flat surface.



### Particle systems

Particle systems can be used for various purposes: for modeling purposes, where they are used to multiply objects according to certain conditions, or for the purposes of physical simulation, which can simulate the most varied effects such as explosions, precipitation and more. many other things.

### Dynamic particle systems

Particle systems can be used to dynamically generate new objects on the surface of selected objects in the scene. The surface of any object can be used as the source of new particles. In the example on the right, on the surface of the sphere, we establish new specimens of the monkey head object. New heads appear on the surface of the sphere (we can choose either in the corners or on the cheeks or just inside the entire volume) and start moving at an initial speed according to the laws of physics, which in this case means that they follow gravity.







In a similar way, you could create particles from an object flying at high speed and simulate an explosion in this way.

#### Use in modeling

As an example of the use of particle systems in modeling, they can be used to arrange repetitive objects on the surface of other objects as shown in the case of spines on the surface of a cactus in the figure below.

We used a system of hair-type particles, where we arranged the hair on selected cactus nodes. These nodes were added to the same group of nodes, which we then used to arrange the spines. For the spike, we modeled a simple cone and used it in the particle system as an object representing individual hair. We added random rotations relative to normal and random sizes above the spines.

### Smoke

The effect of smoke in Blender is basically determined by two objects: (1)

the object of origin, which determines the area from which the smoke will emanate, and (2) the object of the smoke domain, which determines the area where the simulation will be calculated.

The source can be determined by volume, density and temperature difference. Smoke domains, however, can determine the accuracy of the calculation, the density of the simulation. We can also determine smoke parameters such as ignition rate, etc. The image below shows a simple example of smoke and smoke with fire, where the origin is the head of a monkey and the smoke domain is a cuboid.



. . .







# Lecture 09 Animation

### Overview

- 4. What is animation?
- 5. Animation basics
- 6. Animation curves
- 7. Animation pipeline

# What is animation?

Animation is a technique of creating consecutive images or positions of objects in a scene with which we achieve the illusion of continuous movement during the playback of the entire sequence. Animation was basically developed as a sequence of 2D images that represent the temporal sequence of the state of objects in an image. The same technique was later transferred to the 3D environment.

In most cases, the animation does not imitate the real world, but emphasizes certain aspects and thus further illustrates and emphasizes the actions that the characters are supposed to perform in the world.

The 12 basic principles that a good animation must contain are most often highlighted:

- 1. Sqash and stretch is the principle with which we additionally emphasize the deformation of bodies in contact with solid surfaces;
- 2. Aniticipation emphasizes the time when the object / individual is preparing to carry out an action;
- 3. Staging the individual performs actions so that viewers see them;
- 4. Straight ahead action and pose to pose are two different approaches to creating animation. While in the first case we start from the initial state and develop it over time, in the second case we determine the transition between the key states of objects / individuals;
- 5. Follow through and overlapping actions are actions after the completed action and overlapping actions that are performed simultaneously;
- 6. Slow in and slow out all actions have a period of time when they go from sleep to motion and then stop again. These time periods are further emphasized;





- 7. Arc motion most real-world movements are illustrated as curve motion, rarely do objects / specimens move completely straight in a certain direction;
- 8. Secondary actions are actions that add additional details to the primary animation and enliven them;
- 9. Timing timing of the beginnings, ends and overlaps of individual actions and thus their animations;
- 10. Exaggeration often emphasizes the purpose of an individual's action in a particular scene;
- 11. Solid drawing it is important to keep in mind that the specimens are in the 3D space and to adapt them accordingly. This principle is greatly facilitated when the animation itself is actually made in the 3D world;
- 12. Appeal charisma individuals / objects should have charisma and try to attract the viewer's attention in this way.

# Animation basics

It makes sense to take into account the principles presented in the introduction when starting to create your own animation in the Blender tool. Many of the principles listed, of course, do not make sense in cases where we want to recreate what is happening in the real world in a completely realistic way, or perhaps even simulate it. Nevertheless, the listed principles also come into play in such cases as, for example, to animate the view, move the camera and switch between the displayed events.

Blender has a customized user interface for animation purposes that already contains the most commonly used views. To switch to it, select Animation in the information bar instead of the default view. This divides the window as shown in the figure below.







The individual sections are:

- Display of key animation frames (Dope Sheet), where all key animation frames for the selected object (s) are displayed;
- Curves (Graph Editor), which shows how the animated values of the selected object or. facilities change by jumping time;
- 3D view where we have a standard view of the scene;
- Preview through the camera;
- Timeline.

Blender animation is based on the use of key frames, which means that we have to determine which value takes which value at selected times.

## Animation curves

Animation curves determine how animated values change between key frames (be it position, rotation, or something else). Basically, the B-Splines are based on Bezier curves. The transition curves can be edited in the F-Curve view, where we can change the shape of the curve for an individual value and shift its key nodes.

Between individual nodes, instead of B-splices, we can also use another method of interpolation:

- 1. constant interpolation discretely switches between individual values of key frames;
- 2. linear interpolation performs linear interpolation between values;
- 3. Bezier interpolation defines the Bezier curve between values;
- 4. Sinusoidal, square, cubic, quartic, quintic, exponential or circular approximation of values;
- 5. back, bounce or elastic extrapolation of values over certain values.



# Animation pipeline

The basic approach to animation consists of the following steps:

- 1. determine the key positions of objects on the timeline;
- 2. add dependent levels of animations that upgrade the basic animation;
- 3. determine the ways of transitions between individual states and adjust the curves;
- 4. we coordinate individual animations over time.

## How to learn animation?

With a lot of practice. If we have already emphasized this in 3D modeling, this is all the more true for animation. In order to achieve the correct chronology of actions and coordinate them with each other, a lot of practice and comparisons with reference shots are required. Even with a simple bounce of the ball from the ground, we will need quite some time to meet all 12 principles (or at least most that make sense for a given case), and even more so for more complex animations. Try e.g., make a simple walk cycle animation.

. . .





# Lection 10 Render to video

### Overview

1. Video rendering

# Video rendering

The rendering of the videos is related to the fact that we have animation in the scene. It is pointless to portray a stationary scene in a video. For video rendering, you need to set the start and end frames in the rendering settings. As with image rendering, you need to select the resolution, calculator, and other details. In addition, it is necessary to choose to use video format for the format and not image format. At this point, it should be noted that it is possible to replace the default gray background of the image with some other color or with

Most often, FFmpeg video is used to represent the video, for which we can additionally set the encoding parameters in the Encoding section. By default, the Matroska container, H.264 medium output quality and speed format with key frames every 18 frames is used.

For better output, we can further adjust these parameters and thus improve the quality of the rendered video. By default, the video is rendered in a temporary folder, which it makes sense to replace with the folder where we want to save the video.

. . .