Blender is a free and open-source 3D computer graphics software tool set used for creating animated films, visual effects, art, 3D-printed models, motion graphics, interactive 3D applications, virtual reality, and, formerly, video games. Blender's features include 3D modelling, UV mapping, texturing, digital drawing, raster graphics editing, rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animation, match moving, rendering, motion graphics, video editing, and compositing.

Features

Modeling

Forensic facial reconstruction of a mummy by Cícero Moraes

Blender has support for a variety of geometric primitives, including polygon meshes, Bézier curves, NURBS surfaces, metaballs, icospheres, text, and an n-gon modeling system called B-mesh. There is also an advanced polygonal modelling system which can be accessed through an edit mode. It supports features such as extrusion, bevelling, and subdividing.

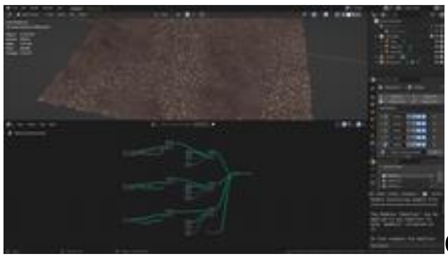
Modifiers

Modifiers apply non-destructive effects which can be applied upon rendering or exporting, such as subdivision surfaces.

Sculpting

Blender has multi-resolution digital sculpting, which includes dynamic topology, "baking", remeshing, re-symmetrization, and decimation. The latter is used to simplify models for exporting purposes (an example being game assets).

Geometry nodes


Geometry Nodes Editor in Blender 3.2

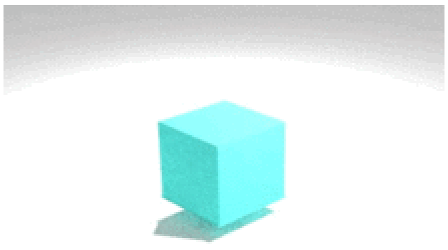
Blender has a geometry node system for procedurally and non-destructively creating and manipulating geometry. It was first added to Blender 2.92, which focuses on object scattering and instancing. It takes the form of a modifier, so it can be stacked over other different modifiers. The system uses object attributes, which can be modified and overridden with string inputs. Attributes can include positions, normals and UV maps. All attributes can be viewed in an attribute spreadsheet editor. The Geometry Nodes utility also has the capability of creating primitive meshes. In Blender 3.0, support for creating and modifying curves objects was added to Geometry Nodes; in the same release, the Geometry Nodes workflow was completely redesigned with fields, in order to make the system more intuitive and work like shader nodes.

Simulation

Cloth simulation

Blender can be used to simulate smoke, rain, dust, cloth, fluids, hair, and rigid bodies.

Fluid simulation



Physics fluid simulation

The fluid simulator can be used for simulating liquids, like water being poured into a cup. It uses Lattice Boltzmann methods (LBM) to simulate fluids and allows for plenty of adjustment of particles and resolution. The particle physics fluid simulation creates particles that follow the smoothed-particle hydrodynamics method.

Blender has simulation tools for soft-body dynamics, including mesh collision detection, LBM fluid dynamics, smoke simulation, Bullet rigid-body dynamics, an ocean generator with waves, a particle system that includes support for particle-based hair, and real-time control during physics simulation and rendering.

In Blender 2.82, a new fluid simulation system called Mantaflow was added, replacing the old FLIP system. In Blender 2.92, another fluid simulation system called APIC, which builds on Mantaflow, was added. Vortices and more stable calculations are improved from the FLIP system.

Cloth Simulation

Cloth simulation is done by simulating vertices with a rigid body simulation. If done on a 3D mesh, it will produce similar effects as the soft body simulation.

Animation

Blender's keyframed animation capabilities include inverse kinematics, armatures, hooks, curve- and lattice-based deformations, shape keys, non-linear animation, constraints, and vertex weighting. In addition, its Grease Pencil tools allow for 2D animation within a full 3D pipeline.

Rendering

An architectural render showing different rendering styles in Blender, including a photorealistic style using Cycles

Blender includes three render engines since version 2.80: EEVEE, Workbench and Cycles.

Cycles is a path tracing render engine. It supports rendering through both the CPU and the GPU. Cycles supports the Open Shading Language since Blender 2.65.

Cycles Hybrid Rendering is possible in Version 2.92 with Optix. Tiles are calculated with GPU in combination with CPU.

EEVEE is a new physically based real-time renderer. While it is capable of driving Blender's real-time viewport for creating assets thanks to its speed, it can also work as a renderer for final frames.

Workbench is a real-time render engine designed for fast rendering during modelling and animation preview. It is not intended for final rendering. Workbench supports assigning colors to objects for visual distinction.

Cycles

Rendering of different materials using the Cycles render engine

Cycles is a path-tracing render engine that is designed to be interactive and easy to use, while still supporting many features. It has been included with Blender since 2011, with the release of Blender 2.61. Cycles supports with AVX, AVX2 and AVX-512 extensions, as well as CPU acceleration in modern hardware.

GPU rendering

Cycles supports GPU rendering, which is used to speed up rendering times. There are three GPU rendering modes: CUDA, which is the preferred method for older Nvidia graphics cards; OptiX, which utilizes the hardware ray-tracing capabilities of Nvidia's Turing architecture & Ampere architecture; and OpenCL, which supports rendering on AMD Radeon graphics cards (with added Intel Iris and Xe support in 2.92). The toolkit software associated with these rendering modes does not come within Blender and needs to be separately installed and configured as per their respective source instructions.

Multiple GPUs are also supported (with the notable exception of the EEVEE render engine) which can be used to create a render farm to speed up rendering by processing frames or tiles in parallel—having multiple GPUs, however, does not increase the available memory since each GPU can only access its own memory. Since Version 2.90, this limitation of SLI cards is broken with Nvidia's NVLink.

Apple's Metal API got initial implementation in Blender 3.1 for Apple computers with M1 chips and AMD graphics cards.

A Benchmark of Blender 3.2 shows great advantages of long supported CUDA against newer API OptiX and brand new HIP for AMD Hardware. Some improvements in performance were added for HIP and OptiX in Blender 3.3 and 3.4.

Support for Intel Arc GPUs arrived in Blender 3.3 LTS.

Supported features

Feature	**CPU**	**CUDA**	**OPTIX**	**HIP**	**oneAPI**	**Metal**

Hardware Minimum for 3.0	x86-64 and other 64-Bit	Cuda 3.0+: Nvidia cards Kepler to Ampere (CUDA Toolkit 11.1+)	OptiX 7.3 with driver 470+: Full: Nvidia RTX Series; Parts: Maxwell+	AMD RDNA architecture or newer, Radeon Software Drivers (Windows, Linux)	Intel Graphics Driver 30.0.101.3430 or newer on Windows, OpenCL runtime 22.10.23904 on Linux	Apple Computers with Apple Silicon in MacOS 12.2, AMD Graphics Cards with MacOS 12.3
Basic shading	Yes	Yes	Yes	Yes	Yes	Yes
Shadows	Yes	Yes	Yes	Yes	Yes	Yes
Motion blur	Yes	Yes	Yes	Yes	Yes	Yes
Hair	Yes	Yes	Yes	Yes	Yes	Yes
Volume	Yes	Yes	Yes	Yes	Yes	Yes
Subsurface scattering	Yes	Yes	Yes	Yes	Yes	Yes
Open Shading Language (1.11) (OSL 1.12.6 in 3.4)	Yes	No	Partial[107]	No	No	No
Correlated multi-jittered sampling	Yes	Yes	Yes	Yes	Yes	Yes
Bevel and AO shaders	Yes	Yes	Yes	Yes	Yes	Yes
Baking	Yes	Yes	Yes	Yes	Yes	Yes
Can use CPU memory		Yes	Yes	Yes	Yes	Yes

Distribute memory across devices	Yes render farm	Yes with NVLink	Yes with NVLink	No	No	No	
Experimental features							
Adaptive subdivision	Experimental	Experimental	Experimental	Experimental	Experimental	Experimental	

Integrator

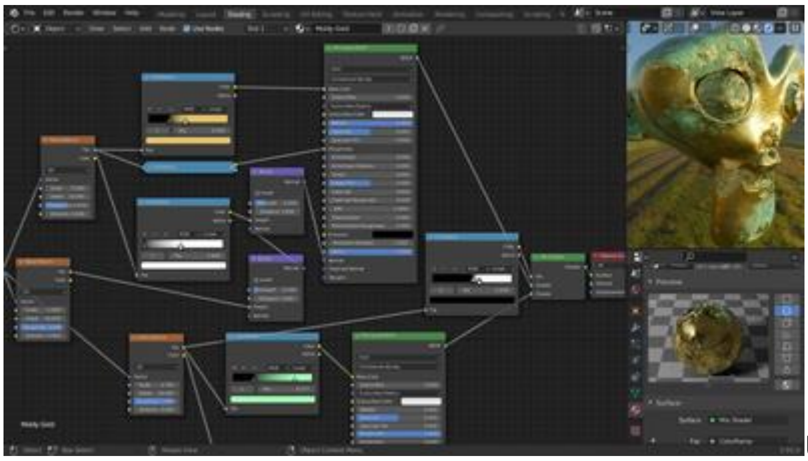
The integrator is the core rendering algorithm used for lighting computations. Cycles currently supports a path tracing integrator with direct light sampling. It works well for a variety of lighting setups, but it is not as suitable for caustics and certain other complex lighting situations. Rays are traced from the camera into the scene, bouncing around until they find a light source (a lamp, an object material emitting light, or the world background), or until they are simply terminated based on the number of maximum bounces determined in the light path settings for the renderer. To find lamps and surfaces emitting light, both indirect light sampling (letting the ray follow the surface bidirectional scattering distribution function, or BSDF) and direct light sampling (picking a light source and tracing a ray towards it) are used.

The default path tracing integrator is a "pure" path tracer. This integrator works by sending several light rays that act as photons from the camera out into the scene. These rays will eventually hit either: a light source, an object, or the world background. If these rays hit an object, they will bounce based on the angle of impact, and continue bouncing until a light source has been reached or until a maximum number of bounces, as determined by the user, which will cause it to terminate and result in a black, unlit pixel. Multiple rays are calculated and averaged out for each pixel, a process known as "sampling". This sampling number is set by the user and greatly affects the final image. Lower sampling often results in more noise and has the potential to create "fireflies" (which are uncharacteristically bright pixels), while higher sampling greatly reduces noise, but also increases render times.

The alternative is a branched path tracing integrator, which works mostly the same way. Branched path tracing splits the light rays at each intersection with an object according to different surface components, and takes all lights into account for shading instead of just one. This added complexity makes computing each ray slower but reduces noise in the render, especially in scenes dominated by direct (one-bounce) lighting.

Open Shading Language

Blender users can create their own nodes using the Open Shading Language (OSL), although it is important to note that this feature is not supported by GPUs.


Using the node editor to create a moldy gold material

Materials

Materials define the look of meshes, NURBS curves, and other geometric objects. They consist of three shaders to define the mesh's surface appearance, volume inside, and surface displacement. The *surface shader* defines the light interaction at the surface of the mesh. One or more bidirectional scattering distribution functions, or BSDFs, can specify if incoming light is reflected, refracted into the mesh, or absorbed. The alpha value is one measure of translucency.

When the surface shader does not reflect or absorb light, it enters the volume (light transmission). If no volume shader is specified, it will pass straight through (or be refracted, see refractive index or IOR) to another side of the mesh.

If one is defined, a *volume shader* describes the light interaction as it passes through the volume of the mesh. Light may be scattered, absorbed, or even emitted at any point in the volume.

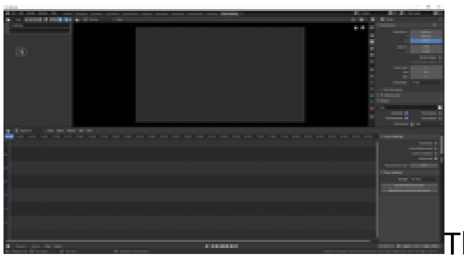
The shape of the surface may be altered by *displacement shaders*. In this way, textures can be used to make the mesh surface more detailed.

Depending on the settings, the displacement may be virtual-only modifying the surface normals to give the impression of displacement (also known as bump mapping) – real, or a combination of real displacement with bump mapping.

Texturing and shading

Blender allows procedural and node-based textures, as well as texture painting, projective painting, vertex painting, weight painting and dynamic painting.

Post-production


The Video Sequence Editor (VSE)

Blender has a node-based compositor within the rendering pipeline, which is accelerated with OpenCL. It also includes a non-linear video editor called the Video Sequence Editor (VSE), with support for effects like Gaussian blur, color grading, fade and wipe transitions, and other video transformations. However, there is no built-in multi-core support for rendering video with the VSE.

Plugins/addons and scripts

Blender supports Python scripting for the creation of custom tools, prototyping, importing/exporting from other formats, and task automation. This allows for integration with several external render engines through plugins/addons. Blender itself can also be compiled & imported as a python library for further automation and development.

File format

Blender features an internal file system that can pack multiple scenes into a single ".blend" file.

- Most of Blender's ".blend" files are forward, backward, and cross-platform compatible with other versions of Blender, with the following exceptions:
  - Loading animations stored in post-2.5 files in Blender pre-2.5. This is due to the reworked animation subsystem introduced in Blender 2.5 being inherently incompatible with older versions.
  - Loading meshes stored in post 2.63. This is due to the introduction of BMesh, a more versatile mesh format.
  - Blender 2.8 ".blend" files are no longer fully backward compatible, causing errors when opened in previous versions.
  - Many 3.x ".blend" files are not completely backwards-compatible as well, and may cause errors with previous versions.
- All scenes, objects, materials, textures, sounds, images, and post-production effects for an entire animation can be packaged and stored in a single ".blend" file. Data loaded from external sources, such as images and sounds, can also be stored externally and referenced through either an absolute or relative file path. Likewise, ".blend" files themselves can also be used as libraries of Blender assets.
- Interface configurations are retained in ".blend" files.

A wide variety of import/export scripts that extend Blender capabilities (accessing the object data via an internal API) make it possible to interoperate with other 3D tools.

Blender organizes data as various kinds of "data blocks" (akin to glTF), such as Objects, Meshes, Lamps, Scenes, Materials, Images, and so on. An object in Blender consists of multiple data blocks – for example, what the user would describe as a polygon mesh consists of at least an Object and a Mesh data block, and usually also a Material and many more, linked together. This allows various data blocks to refer to each other. There may be, for example, multiple Objects that refer to the same Mesh, and making subsequent editing of the shared mesh results in shape changes in all Objects using this Mesh. Objects, meshes, materials, textures, etc. can also be linked to other .blend files, which is what allows the use of .blend files as reusable resource libraries.

Import and export

The software supports a variety of 3D file formats for import and export, among them Alembic, 3D Studio (3DS), FBX, DXF, SVG, STL (for 3D printing), UDIM, USD, VRML, WebM, X3D and OBJ.

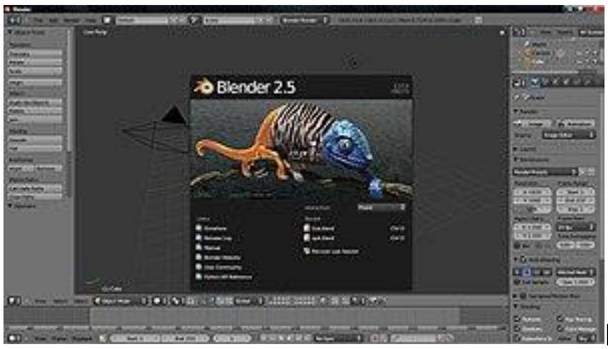
Blender Game Engine

The Blender Game Engine was a built-in real-time graphics and logic engine with features such as collision detection, a dynamics engine, and programmable logic. It also allowed the creation of stand-alone, real-time applications ranging from architectural visualization to video games. In April 2018, the engine was removed from the upcoming Blender 2.8 release series, due to updates and revisions to the engine lagging behind other game engines such as Unity and the open-source Godot. In the 2.8 announcements, the Blender team specifically mentioned the Godot engine as a suitable replacement for migrating Blender Game Engine users.

Blender Internal

Blender Internal, a biased rasterization engine and scanline renderer used in previous versions of Blender, was also removed for the 2.80 release in favor of the new "EEVEE" renderer, a realtime physically based renderer.

User interface



Blender's user interface underwent a significant update with Blender 2.57, and again with the release of Blender 2.80.

Commands

Most of the commands are accessible via hotkeys. There are also comprehensive graphical menus. Numeric buttons can be "dragged" to change their value directly without the need to aim at a particular widget, as well as being set using the keyboard. Both sliders and number buttons can be constrained to various step sizes with modifiers like the `Ctrl` and `Shift` keys. Python expressions can also be typed directly into number entry fields, allowing mathematical expressions to specify values.

Modes

Blender includes many modes for interacting with objects, the two primary ones being *Object Mode* and *Edit Mode*, which are toggled with the `Tab` key. Object mode is used to manipulate individual objects as a unit, while Edit mode is used to manipulate the actual object data. For example, an Object Mode can be used to move, scale, and rotate entire polygon meshes, and Edit Mode can be used to manipulate the individual vertices of a single mesh. There are also several other modes, such as Vertex Paint, Weight Paint, and Sculpt Mode.

Workspaces

The Blender GUI builds its tiled windowing system on top of one or multiple windows provided by the underlying platform. One platform window (often sized to fill the screen) is divided into sections and subsections that can be of any type of Blender's views or window types. The user can define multiple layouts of such Blender windows, called screens, and switch quickly between them by selecting from a menu or with keyboard shortcuts. Each window type's own GUI elements can be controlled with the same tools that manipulate the 3D view. For example, one can zoom in and out of GUI-buttons using similar controls, one zooms in and out in the 3D viewport. The GUI viewport and screen layout are fully user-customizable. It is possible to set up the interface for specific tasks such as video editing or UV mapping or texturing by hiding features not used for the task.

Use in industry

Blender started as an in-house tool for NeoGeo, a Dutch commercial animation company. The first large professional project that used Blender was *Spider-Man 2*, where it was primarily used to create animatics and pre-visualizations for the storyboard department.

The French-language film *Friday or Another Day* (*Vendredi ou un autre jour*) was the first 35 mm feature film to use Blender for all the special effects, made on Linux workstations.[It won a prize at the Locarno International Film Festival. The special effects were by Digital Graphics of Belgium.

Tomm Moore's *The Secret of Kells*, which was partly produced in Blender by the Belgian studio Digital Graphics, has been nominated for an Oscar in the category "Best Animated Feature Film". Blender has also been used for shows on the History Channel, alongside many other professional 3D graphics programs.

Plumíferos, a commercial animated feature film created entirely in Blender, had premiered in February 2010 in Argentina. Its main characters are anthropomorphic talking animals. Blender is used by NASA for many publicly available 3D models. Many 3D models on NASA's 3D resources page are in a native .blend format.

Special effects for episode 6 of *Red Dwarf* season X, screened in 2012, were created using Blender as confirmed by Ben Simonds of Gecko Animation. Blender was used for previsualization in *Captain America: The Winter Soldier*.

Some promotional artwork for *Super Smash Bros. for Nintendo 3DS and Wii U* was partially created using Blender. The alternative hip-hop group Death Grips has used Blender to produce music videos. A screenshot from the program is briefly visible in the music video for *Inanimate Sensation*.

The visual effects for the TV series *The Man in the High Castle* were done in Blender, with some of the particle simulations relegated to Houdini. NASA also used Blender to develop an interactive web application Experience Curiosity to celebrate the 3rd anniversary of the *Curiosity* rover landing on Mars. This app makes it possible to operate the rover, control its cameras and the robotic arm and reproduces some of the prominent events of the Mars Science Laboratory mission. The application was presented at the beginning of the WebGL section on SIGGRAPH 2015.

Blender was used for both CGI and compositing for the movie *Hardcore Henry*. The visual effects in the feature film *Sabogal* were done in Blender. VFX supervisor Bill Westenhofer used Blender to create the character "Murloc" in the 2016 film *Warcraft*.

Director David F. Sandberg used Blender for multiple shots in *Lights Out*, and *Annabelle: Creation*. Blender was used for parts of the credit sequences in *Wonder Woman*[181] Blender was used for doing the animation in the film *Cinderella the Cat*.

VFX Artist Ian Hubert used Blender for the science fiction film Prospect. The 2018 film *Next Gen* was fully created in Blender by Tangent Animation. A team of developers worked on improving Blender for internal use, but it is planned to eventually add those improvements to the official Blender build.

The 2019 film *I Lost My Body* was largely animated using Blender's Grease Pencil tool by drawing over CGI animation allowing for a real sense of camera movement that is harder to achieve in purely traditionally drawn animation.[186] Ubisoft Animation Studio will use Blender to replace its internal content creation software starting in 2020.

Khara and its child company Project Studio Q are trying to replace their main tool, 3ds Max, with Blender. They started "field verification" of Blender during their ongoing production of *Evangelion: 3.0+1.0*. They also signed up as Corporate Silver and Bronze members of Development Fund.

The 2020 film *Wolfwalkers* was partially created using Blender. In 2021 SPA Studios started hiring Blender artists and as of 2022, contributes to Blender Development. The 2021 Netflix production Maya and the Three was created using Blender. Warner Bros. Animation started hiring Blender artists in 2022. VFX company Makuta VFX used Blender for the VFX for Indian blockbuster RRR.

Blender, the versatile open-source 3D software, offers a robust set of **VFX tools**. Let's explore some of its features:

1. **Compositing**:
- Blender includes a fully-fledged built-in **compositor**. This powerful tool allows you to post-produce your renders without leaving Blender.
- Key features of the compositor:
- **Impressive library of nodes**: Use nodes for creating camera effects, color grading, vignettes, and more.
- **Render-layer support**: Seamlessly integrate different render layers.
- **Full compositing**: Combine images and video files.
- **MultiLayer OpenEXR files**: Render to multi-layered OpenEXR files.
- **Multi-threaded**: Efficiently handle complex compositing tasks.

2. **Motion Tracking**:
- Blender's production-ready **camera and object tracking** streamlines your workflow.
- Features include:
- **Automatic and manual tracking**: Track camera movements and object motion.
- **Powerful camera reconstruction**: Reconstruct the camera's position and orientation.
- **Real-time preview**: Visualize tracked footage within your 3D scene.
- **Planar tracking and Tripod solvers**: Handle various tracking scenarios.

3. **Add-ons and Community Tools**:
- Blender's vibrant community contributes various add-ons for enhanced VFX capabilities.
- Some popular add-ons include:
- **Mesh/Curve Library**: Access pre-built meshes and curves.
- **Mesh Tools**: Perform operations like splitting edges, welding vertices, and normal adjustments.
- **UV Checker**: Useful for UV mapping.
- **Export Tools**: Simplify exporting to formats like .fbx.
- **Bulk Export**: Streamline exporting multiple objects.

Remember, Blender's versatility extends beyond VFX—it's also a powerful tool for modeling, animation, and more!

4. **Smoke and Fire Simulation**:
- Blender's **smoke and fire simulation** system allows you to create realistic smoke, fire, and explosions.
- Key features:

- **Smoke Domain**: Define the area where smoke or fire will appear.
- **Smoke Flow**: Create emitters for smoke or fire.
- **High-resolution simulations**: Achieve detailed results.
- **Color and Density Control**: Adjust the appearance of smoke and fire.
- **Noise and Turbulence**: Add realism to the simulations.

5. **Dynamic Paint**:
   - This tool lets you use objects as brushes to paint dynamic effects onto other objects.
   - Examples:
     - **Wet Maps**: Objects can leave wet marks on surfaces.
     - **Displacement Maps**: Objects can displace other surfaces.
     - **Weight Maps**: Influence particle systems or physics simulations.

6. **Grease Pencil**:
   - Blender's **2D animation toolset**, the Grease Pencil, is also useful for VFX.
   - Features:
     - **Frame-by-frame animation**: Create hand-drawn effects.
     - **Strokes and Fills**: Animate strokes and fill areas.
     - **Layer System**: Organize your drawings.
     - **3D Integration**: Combine 2D and 3D elements.

Remember, Blender's flexibility allows you to explore various creative avenues. Whether you're working on visual effects, animation, or modeling, Blender has you covered! 🎨✨