Blender 3D By Example

Second Edition

A project-based guide to learning the latest Blender 3D, EEVEE rendering engine, and Grease Pencil

Oscar Baechler and Xury Greer

Packt
www.packt.com

Blender 3D By Example
Second Edition

A project-based guide to learning the latest Blender 3D, EEVEE rendering engine, and Grease Pencil

Oscar Baechler
Xury Greer

Packt

BIRMINGHAM - MUMBAI

Blender 3D By Example
Second Edition

Commissioning Editor: Kunal Chaudhari
Acquisition Editor: Ashitosh Gupta
Content Development Editor: Akhil Nair
Senior Editor: Hayden Edwards
Technical Editor: Shubham Sharma
Copy Editor: Safis Editing
Project Coordinator: Kinjal Bari
Proofreader: Safis Editing
Indexer: Manju Arasan
Production Designer: Alishon Mendonsa

First published: September 2015
Second edition: May 2020

Production reference: 1280520

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78961-256-1

www.packt.com

Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the authors

Oscar Baechler is a CG generalist, professor, painter, photographer, open source advocate, and community organizer who teaches at Lake Washington Institute of Technology. He's published a number of mobile games with a Blender pipeline and created animation for clients both big and small. Oscar runs the Seattle Blender User Group and Ballard Life Drawing Co-op and has presented on CGI at SIGGRAPH, LinuxFest Northwest, the Blender Conference, OSCON, Usenix LISA, SeaGL, SIX, WACC, and others.

I want to thank my wife Roxanne, for her endless support; my kids Felix and Susan, for playing Baby Dragon with me; and my mom and dad for giving me art supplies. Also, I want to thank my friend Jacob, for his scanning help. Thanks to Lake Washington Institute of Technology, and all the students who have shared the classroom with me. Thank you Xury, for writing a book with me! Lastly, thanks to the Blender community, especially the Seattle Blender User Group, for over a decade of Blender and friendship.

Xury Greer has been involved in digital media production for over 15 years. He got his start as an indy film director, participating in 48-hour film competitions, and creating training videos for businesses in the Greater Seattle Area. Xury earned his bachelor's degree in game design at Lake Washington Institute of Technology and graduated with the highest honors. Xury specializes in 3D characters and technical art, and he loves to share his knowledge. He has taught courses for Mount Si High School, Washington Network for Innovative Careers, DigiPen, and LWTech.

About the reviewers

Henk Kok is an experienced 3D generalist with over a decade of experience working on games, television series, and feature films. He values thriving cooperation with those around him. In 2019 he worked as the 3D animation supervisor for the groundbreaking Amazon Prime series *Undone*.

Fernando Castilhos Melo lives in Toronto, Canada, and works as a software engineer. He holds a degree in computer science. In his spare time, he works on 3D modeling using Blender and has done so since 2009. He has given some lectures about Blender and 3D modeling at some open source software events and reviewed several Blender books. He also developed an integration between Blender and Kinect to create 3D animation using body movements.

I would like to say a big thank you to:
-My wife, Mauren, for all the support
-My parents, Eloir and Miriam, for encouraging me
-My dog, Polly, for being (literally) at my side all the time during this review
-All my friends for giving me the confidence for this work

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface

What is Blender? In a nutshell, it's a free, open source 3D modeling suite. But it's also a 3D and 2D animation program. But wait! It's also a video editor. And a Python programming IDE. And a sculpting interface, a compositor, a motion tracker, and so much more. A nutshell is insufficient; Blender needs more of a watermelon to cover all its features.

I remember my first experience with Blender. After years with the commercial program Maya, I was skeptical of how free software could compare, and learning 3D programs is rarely forgiving to newcomers. But every day I used Blender, I would discover an awesome killer feature Blender had that expensive commercial 3D animation software didn't. Then a new update for Blender would launch, and the new features and workflows would leave me floored again.

Over the years, I've met so many Blender users who retell this story in their own words. The conclusion is generally the same: we can't believe how lucky we are that such phenomenal software exists, free for everyone, and supported by a robust open source community. The energy is infectious, and long-time Blender users talk about this magic piece of software with a passion usually reserved for wedding speeches.

We look forward to introducing you to Blender in this book. The chapters we've put together will help Blender beginners leap over the initial stumbling blocks of Blender's tools and interface. They'll also challenge your artistic and technical skills with advanced workflows to bring your imagination to life.

Who this book is for

This book assumes that you have a decent familiarity with computers. You should have some rudimentary skills at using a mouse and keyboard, navigating the internet, and know your way around your computer's operating system. You'll also need some grasp of your hardware. Tools such as a keyboard with a number pad, a mouse with three buttons, and a powerful CPU and GPU are helpful, but you can make some decent Blender projects using even a bare bones netbook.

Maybe you have some knowledge of other 3D programs, such as ZBrush, Maya, Cinema4D, or SketchUp. Maybe you're looking to learn Blender because the price tag appeals, the newest features blow you away, or a professional client wants you working with their Blender-centric pipeline. You'll find many concepts that have equivalencies to those programs. You can gloss over the explanations of 3D vocabulary, such as polygons, normals, and rigs, and instead jump right into Blender's tools and workflows.

For those of you who enjoy gaming, that experience will also pay off when learning Blender. Just like many AAA games, Blender's interface is heavily based on keyboard shortcuts, and navigating the 3D viewport is similar to moving the view around in games. Additionally, Blender is an excellent choice for any gamers looking to create content for their favorite game's custom mod community.

If you're more inclined toward coding, you might have experience with a game engine, such as Unity, Unreal, or Godot. This book will serve as an excellent expansion to your technical skills, showing you how to make 3D content to add to games. This book will also give you a foundation in understanding how these 3D components work, which will come in handy if you ever build out your own 3D tools.

Artists of every stripe can find their new favorite tool in Blender. Blender's 2D animation tools are perfect for artists who love drawing, anime, and experimental media. Especially when combined with Blender's native 3D interface, plus a drawing tablet, Blender has everything you need for both 3D and hand-drawn cartoons.

Whether you're completely new to Blender, or a 3D animation veteran enticed by Blender's newest features, this book will have something for you.

What this book covers

`Chapter 1`, *Introduction to 3D and the Blender User Interface*, explains the basics of Blender's interface, tools, and workflow conventions.

`Chapter 2`, *Editing a Viking Scene with a Basic 3D Workflow*, will take a look at a 3D scene and let us get used to navigating and transforming objects.

`Chapter 3`, *Modeling a Time Machine – Part 1*, is the beginning of a two-part project in which we will model an object based on provided reference images. We will cover many of the essential modeling tools needed for creating 3D objects.

`Chapter 4`, *Modeling a Time Machine – Part 2*, is the second half of the time machine project. We will build on our modeling knowledge and discover non-destructive workflows.

`Chapter 5`, *Modern Kitchen – Part 1: Kitbashing*, will show how to plan a complete scene and model the necessary assets to complete a kitchen layout.

`Chapter 6`, *Modern Kitchen – Part 2: Materials and Textures*, is a deep dive into material nodes and explains how to create all kinds of materials to decorate our kitchen with.

`Chapter 7`, *Modern Kitchen – Part 3: Lighting and Rendering*, is the final chapter in the kitchen series. We will produce a final rendered image complete with lighting and post-processing effects.

`Chapter 8`, *Illustrating an Alien Hero with Grease Pencil*, is the first of our three chapters that dives into the brand new feature set known as Grease Pencil. We will learn about character concept art workflows and how to use the basics of Grease Pencil.

`Chapter 9`, *Animating an Exquisite Corpse in Grease Pencil*, builds on the previous chapter's workflows and dives into animation and key frames with a loose and fun animation style.

`Chapter 10`, *Animating a Stylish Short with Grease Pencil*, wraps up the Grease Pencil projects in this book. We will cover more advanced workflows and explain how to animate something with more structure than the previous chapter.

`Chapter 11`, *Creating a Baby Dragon – Part 1: Sculpting*, is the beginning of the biggest project in this book. We'll start with an introduction to sculpting. We'll see an overview of the brushes and learn how to create our very own baby dragon design, which we will take all the way to a game-ready asset by the final chapter.

`Chapter 12`, *Creating a Baby Dragon – Part 2: Retopology*, is a shift into the more technical side of 3D character creation. We'll learn about shrink-wrapping, surface snapping, and rules of topology to transform the sculpted baby dragon into a low-poly mesh that can be used in a production pipeline.

`Chapter 13`, *Creating a Baby Dragon – Part 3: UV Unwrapping*, is where we'll prepare the model for texture painting. We'll learn how to cut seams, unwrap UVs, lay out islands, and use checker patterns to check for distortion.

`Chapter 14`, *Creating a Baby Dragon – Part 4: Baking and Painting Textures*, gets back to the artistic side of things. We'll start by baking texture maps that can be used as masks in our texture painting workflow. We'll use Blender's built-in texture painting tools, and we'll add some color and surface detail to the baby dragon.

`Chapter 15`, *Creating a Baby Dragon – Part 5: Rigging and Animation,* is the final baby dragon chapter. We'll get to see all of our hard work pay off and rig the dragon so that it can be posed and animated. To wrap it up, we'll animate a fly cycle so that we can see the character in action.

`Chapter 16`, *The Wide World of Blender*, shows off some of the areas that this book couldn't cover in detail. Even a book this size can barely scratch the surface of what a 3D suite such as Blender can do, but we'll have a look at some inspiring extra features before we're done.

To get the most out of this book

You will need an internet connection to download the latest version of Blender and the source files for this project (an internet connection is not required after downloading the software and the files).

Blender can be downloaded from `https://www.blender.org/download` and requires about 400 MB of storage space to install. The project files in this book are approximately 2.65 GB all together (you do not need to download them all at once). There are some projects that require additional software for digital painting / image editing such as Krita, GIMP, Affinity Photo, or Photoshop. We recommend Krita because it's free and open source, just like Blender! It can be downloaded from `https://krita.org/en/download/krita-desktop/`. The download for Krita is approximately 100 MB. It's a good idea to have at least 4 GB of additional free storage so that you can create your own 3D sculptures and texture files for the projects. In total, about 7 GB of space will be enough for everything covered in this book.

The version of Blender used in this book requires a computer that supports OpenGL 3.3. You can find the official hardware requirements on the Blender website here: `https://www.blender.org/download/requirements/` This book has been tested for the Blender 2.8 series of releases. The upcoming 2.9 and 3.0 series of releases may have different hardware requirements.

Blender works best on a computer that has a numpad because the camera navigation hotkeys are bound the numpad keys. This book uses an alternative method to teach these controls since many laptops do not have a numpad, so it is not absolutely required.

A drawing tablet with pressure sensitivity is highly recommended for the Grease Pencil projects as well as the sculpting and texturing chapters in the Baby Dragon project. There are affordable options available from `http://www.huion.com/`, premium options on offer from `https://www.wacom.com/`, or if you have a computer with a built-in stylus such as a Microsoft Surface that will work nicely as well. It is possible to complete these chapters using a mouse, but it is not recommended.

Software/Hardware covered in the book	OS requirements
Blender 2.83	Windows 10, 8.1, and 7 macOS 10.12+ Linux
Krita 4.2.9	Windows 8.1 or higher, OSX 10.12, Linux

Blender and Krita are open source and receive updates several times a year. There may be new versions available if you're picking up this book even a few months after it's published, but don't worry. The projects in this book should still be compatible.

Download the example code files

You can download the example code files for this book from your account at `www.packt.com`. If you purchased this book elsewhere, you can visit `www.packtpub.com/support` and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at `www.packt.com`.
2. Select the **Support** tab.
3. Click on **Code Downloads**.
4. Enter the name of the book in the **Search** box and follow the onscreen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR/7-Zip for Windows
- Zipeg/iZip/UnRarX for Mac
- 7-Zip/PeaZip for Linux

The project files for the book are also hosted on GitHub at `https://github.com/PacktPublishing/Blender-3D-By-Example-Second-Edition`. In case there's an update to the project files, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://static.packt-cdn.com/downloads/9781789612561_ColorImages.pdf`.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "Rename it `Top`."

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Choose **Delete | Faces**."

Warnings or important notes appear like this.

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at `customercare@packtpub.com`.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit `packt.com`.

1
Introduction to 3D and the Blender User Interface

Welcome to the wonderful world of 3D graphics! This section of this book will help you jump-start your knowledge with some terminology and the basics of working in 3D. We'll keep this brief and try to get through the boring stuff as quickly as possible so you can get right into creating amazing 3D projects in Blender 2.8!

Blender 2.8 is a series of releases. There is usually an update for the software every 3 to 4 months. The first release in the series was 2.80, then 2.81, 2.82, and so on. The projects in this book can be completed with version 2.80 onward, with some optional features requiring 2.81 onward. You can read more about Blender's release cycle here: `blender.org/download/releases`.

First, we will take a look at the fundamentals of a 3D scene. We will learn how the 3D coordinate system uses three dimensional axes, as well as how 3D objects are manipulated with transformations. We will answer some basic questions, such as: what are objects? What are polygons? What is topology? What are materials and textures? What is the difference between Perspective and Orthographic views? The answers to these questions are key to working with any 3D software.

After we've provided you with some general 3D knowledge, we will learn about the specifics of Blender. We will cover how to install the software, as well as how to download the source files for this book. We will take a look at Blender 2.8's user interface. Then, we will learn about the basic 3D navigation controls, which include Rotate, Zoom, and Pan. We will also learn how to use Blender's hotkeys effectively. At the end of this chapter, we will provide an overview of the projects in this book.

We will cover the following topics in this chapter:

- Overview of the 3D workflow
- Blender 2.8's user interface
- Basic 3D navigation controls
- A brief introduction to the projects in this book

Overview of the 3D workflow

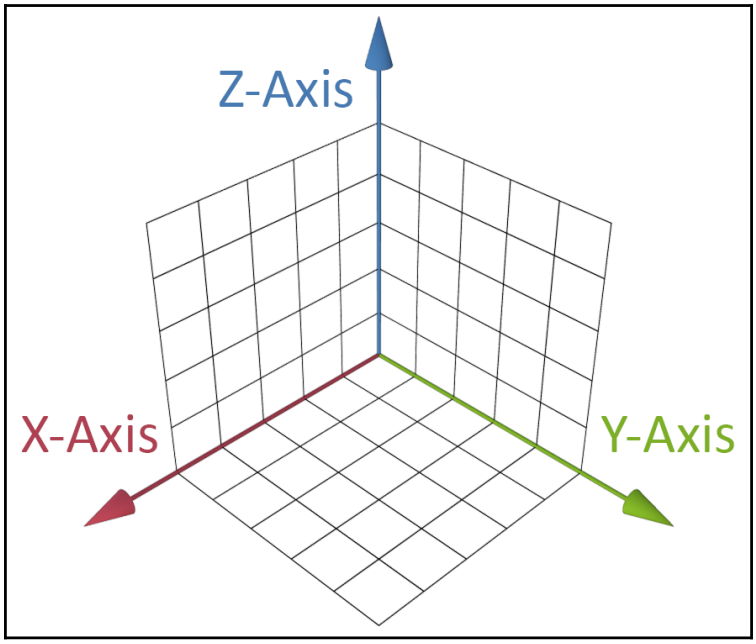
If this is your first time working with 3D software, you'll find the explanations in this section very helpful. However, if you are already familiar with 3D terminology and the composition of a 3D scene, then you may want to skip ahead to the next section of this chapter.

Some of the vocabulary terms you're about to learn might sound overwhelming at first, but don't worry – you don't have to be good at math just because we use words such as "geometry" to describe our 3D models. Luckily for us, the software does all of the complex math for us, and we get to sit back and create art without having to worry about it – hooray!

The 3D coordinate system

All 3D software uses the Cartesian coordinates system, which is made up of three-dimensional axes: the ***X*-Axis** (red), the ***Y*-Axis** (green), and the ***Z*-Axis** (blue). The exact unit size of this coordinate system is arbitrary and varies from one software package to another, but many packages set one unit on the grid to be equal to 1 meter in the real world:

There is a special type of 3D software known as **Computer-Aided Design** (**CAD**). This is used for engineering and conforms more closely to real-life units, but for the purposes of this book, we will not be discussing CAD software.



The three-dimensional axes: **X-Axis** (red), **Y-Axis** (green), and **Z-Axis** (blue)

With these three axes, we can define where an object is in a 3D space using **transforms**. There are three types of transforms:

- **Location**: (sometimes called **translation**) This determines the position of an object.
- **Rotation**: This determines the orientation of an object.
- **Scale**: This determines the size of an object.

Now that we understand the coordinate system, let's look at the 3D objects that will appear in the scene.

3D objects

An object is something that appears in a 3D scene. All objects have transforms that define their location, rotation, and scale in a 3D space. You will find several types of objects in a 3D scene:

- **Mesh**: A mesh is the most common type of object in 3D; nearly everything we make is a mesh. Meshes are 3D objects that are made up of components (sometimes referred to as the geometry of the mesh). These components are used to form geometric polygons. Polygons are the multi-sided shapes that form the visible surface of a model. Creating 3D models with this approach is called polygonal modeling.
- **Empty**: An empty is an object that doesn't have any components attached to it. Some software packages call these null objects or locators. These are useful in advanced workflows for defining and keeping track of an exact spot in a 3D space. Since an empty has transforms, it will be present in the 3D scene just like all other objects, but because it has no components, it will not be visible in the final result.
- **Light**: A light is a type of object that casts light onto the scene. Just like in the real world, you can't see without a light source. If a 3D scene had no light source, you would just see black. Most 3D software includes a light source in the scene by default so that you can see what you're doing. Often, these default lights are a type of environmental light or ambient light source that illuminates the scene without necessarily coming from a particular point in the scene.
- **Camera**: A camera is a tool that's used to create the final image from our 3D scene. We can use a 3D camera the same way we would use a camera in real life: position it, aim it at the subject, and take a picture. The picture we take with a 3D camera is called a render. Rendering creates a high-quality image of the scene. High-quality renders take much longer to process than the normal Viewport preview of the scene, so we don't usually render until we are finished creating the scene.

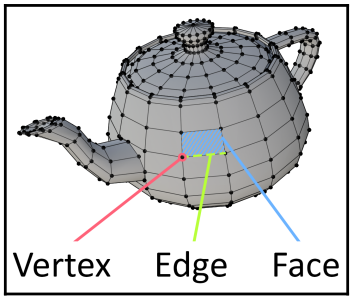
Now that we know what an object is, let's take a closer look at the most important type of object: a mesh. We need to understand how the components of a mesh come together to create a 3D model.

Components of a mesh

There are three basic components that we use in polygonal modeling:

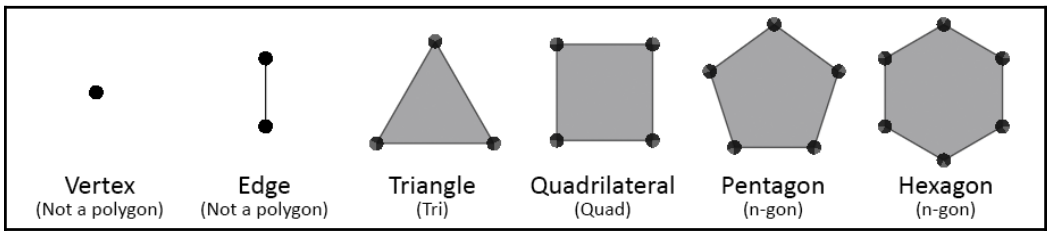
- **Vertices**: The most basic piece of geometry is a vertex (the plural form is vertices). A vertex is a single point in 3D space. It has no size nor orientation; it only has a location within the mesh object. You can't do much with vertices alone, which is why we need edges**.**
- **Edges**: These are straight lines that are drawn between vertices, similar to a connect-the-dots puzzle. The edges that connect two points are always perfectly straight in polygonal modeling.
- **Faces**: The visible part of a polygon. Faces are created by filling in the space between three or more edges.

The following diagram shows the vertices, edges, and faces of a 3D model:



The three basic components of a mesh

Polygons can have any number of sides; three sides make up a **triangle** (**tri**), while four sides make up a **quadrilateral** (**quad**). There are lots of fancy names for specific polygons with more than four sides, such as pentagon, hexagon, and so on, but in the world of 3D modeling, any polygon with more than four sides is simply referred to as an **n-gon**. The following image shows some of the basic polygons you'll come across:

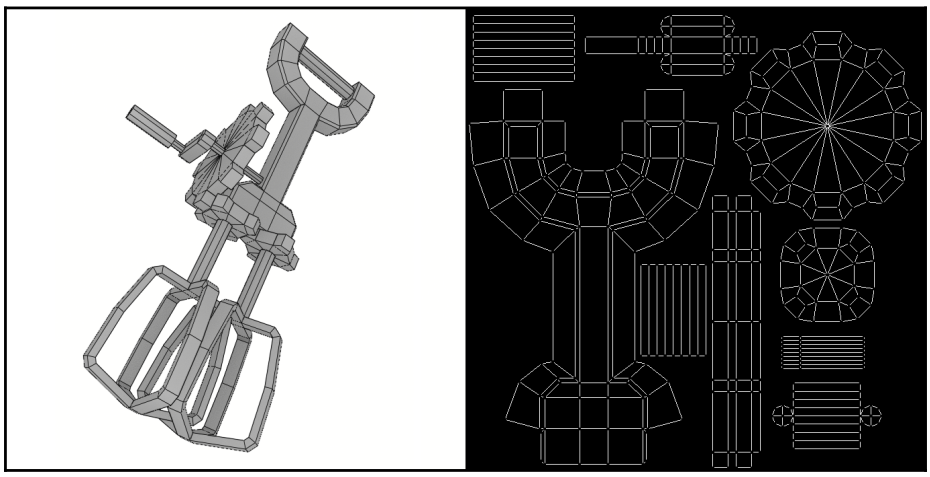


Vertex, edge, tri, quad, and n-gons

The way in which these components are connected is referred to as **topology**, a subject that we will cover in depth later in this book. There are many best practices and rules for creating a mesh with good topology. The most basic rule of topology is that quadrilaterals are the best type of polygon, triangles should be used sparingly, and n-gons should be avoided altogether. Models that don't follow the rules of good topology usually have problems in the final result. Topology is a very large and advanced subject, so we won't go into any more detail about it in this chapter.

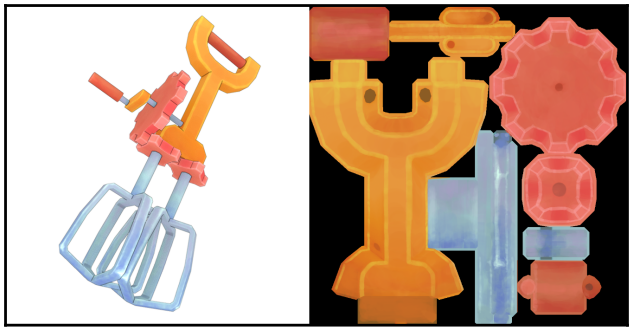
Materials and textures

We can add color to our 3D models with a mixture of materials and textures. Materials are used to determine how light behaves when it interacts with the surface of the object. Does it look like glass? Metal? Skin? Textures are 2D images that are wrapped onto a 3D model, sort of like a candy wrapper. To make our textures line up with the model, we have to unwrap the model first. Unwrapping gives us a 2D representation of the model called a UV map (or UVs). An example of this can be seen in the following image:



The 3D model on the left has been unwrapped to create the UVs on the right.

Once we have UVs, we can paint a texture that will be wrapped back onto the model, as shown here:

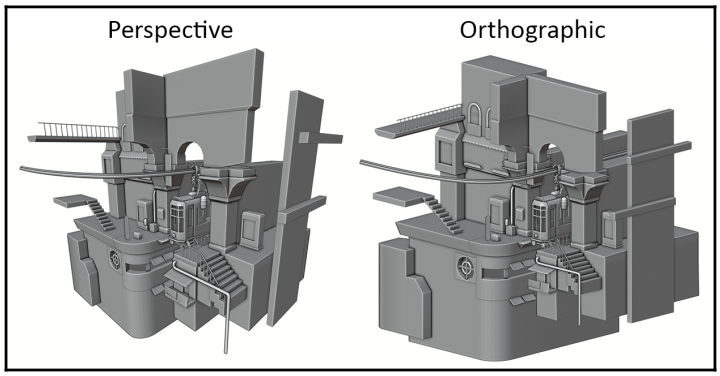


The 3D model on the left has been given a texture from the 2D image on the right

They are called UVs because all of the pieces have a U coordinate and a V coordinate, which are used to determine their positions in 2D space (very similar to graphing data on a 2D graph). Since we already used *X*, *Y*, and *Z* for our three-dimensional axes, our two-dimensional axes are labeled *U* and *V*. We will cover UVs, materials, and textures in detail in later chapters.

Perspective view versus Orthographic view

3D scenes can be displayed in Perspective mode or in Orthographic mode. In Perspective mode, objects are drawn with a vanishing point. As objects get farther away from us, they look smaller, which is the way things look in real life. In Orthographic mode, however, objects stay the same size no matter how far away they are from us. In this mode, everything looks flat and close together. This can be useful for making blueprints or architectural renders, but usually, we keep the view in Perspective mode because it looks more natural:



Perspective versus Orthographic

So there it is – your first introduction to 3D! We've covered a lot of new ideas in a short time, but they will all become second nature to you once you've spent a little time working on 3D projects. Next, we'll take a look at Blender's user interface.

Blender's user interface

You can download the latest version of Blender at `http://blender.org/download`. At the time of writing, the latest version is Blender 2.83. Blender is available on Windows, macOS, and Linux. It is a very similar experience on all three operating systems.

Most of Blender's hotkeys are the same between operating systems. However, if you are following along with this book using a Mac, you need to use the *command* (*cmd*) key instead of the *control* (*ctrl*) key any time the instructions say to use the control key.

Before we learn about the current version of the user interface, it's useful to know a little bit about Blender's history. There is over a decade's worth of tutorials and resources available online. The software may look a bit different in those old resources, but if you can get past the old interface, the information is just as helpful as it always was, so let's take a look.

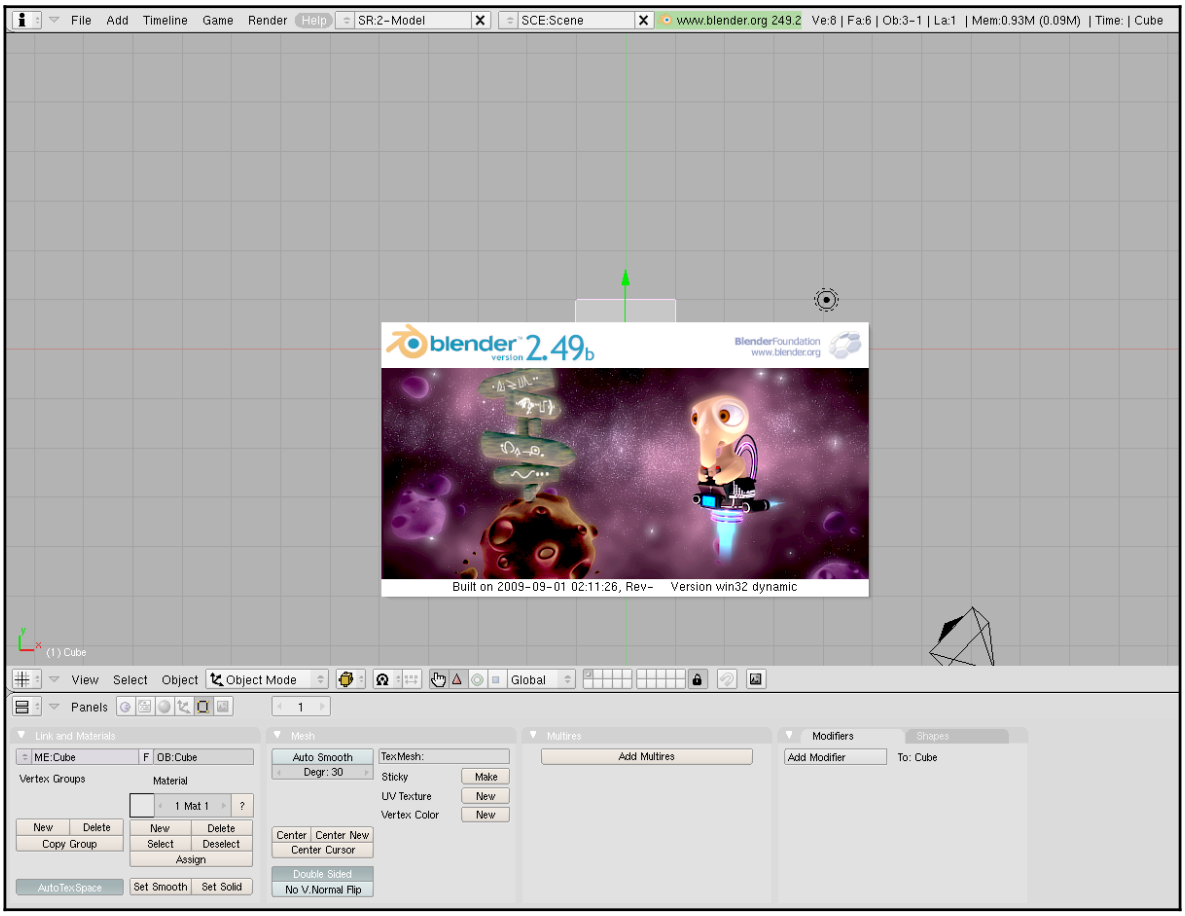
A brief history of Blender's user interface

Blender's **user interface** (**UI**) was very polarizing in the past. Older versions of Blender required the user to memorize dozens of hotkeys before it was possible to accomplish even basic tasks, which meant that many users found this hard to use. However, those were the days of Blender 2.49, and when Blender was updated to version 2.50, the UI got its first major facelift, which added many new features, more buttons, and a cleaner user experience.

Each release of Blender increments the version number by +0.01, which means 2.80 is 30 versions newer than 2.50 – that's a lot of versions! Many of these versions simply added small new features and bug fixes, but version 2.80 is just as big of an overhaul from version 2.79 as 2.50 was from version 2.49.

The original Blender included most of the basic requirements for a 3D modeling suite: 3D modeling, rigging, animating, and its internal "Blender Render" rendering engine. The earliest versions of the software were infamous for missing features such as undo and warning the user that data might be lost if they exited the program without saving first.

Its bright white UI with horizontal buttons, excessive use of tabs, and odd coloration was criticized by many users, but nevertheless, it was responsible for some amazing Open Movie projects such as "Elephant's Dream" and "Big Buck Bunny":



The UI for Blender 2.49

When Blender 2.50 rolled around, the UI was changed radically in response to user feedback. The Open Movie "Sintel" was created alongside the development of this new version to make sure that it included all of the features required for animation production. The 2.5 series of releases continued to introduce amazing features such as Cycles, the ray tracing rendering engine; the bMesh modeling system, which overhauled all of the modeling tools and allowed users to use n-gons; the new dynamic topology sculpting tools; and much more.