

# CHAPTER 1

---

## Introduction

Virtual reality is a medium with tremendous potential. The ability to be transported to other places, to be fully immersed in experiences, and to feel like you're really there—*present*—opens up previously unimagined ways to interact and communicate. Until recently, virtual reality was out of reach for the average consumer due to cost and other factors. However, advances in the technology over the last few years have set the stage for a mass-market revolution that could be as influential as the introduction of television, the Internet, or the smartphone.

Virtual reality—VR for short—comprises a collection of technologies: 3D displays, motion tracking hardware, input devices, software frameworks, and development tools. While consumer-grade VR hardware is young and evolving, a handful of platforms have emerged as go-to choices, including the Oculus Rift, Samsung Gear VR, and Google Cardboard. Each delivers a different level of VR experience, at a different price point, with varying degrees of in-your-hands portability.

Software to create and display consumer virtual reality is also coming together rapidly. The Unity3D and Unreal game engines, popular for making desktop and mobile games, have become tools of choice for native VR development. And the Web is not far behind: WebGL and 3D JavaScript frameworks like Three.js and Babylon.js are providing a path for creating open source, browser-based virtual reality experiences for desktop and mobile operating systems.

It's an exciting time! With so much energy going into development, and so much consumer interest, VR just might be the next big wave of computer technology. In this book, we explore the hardware, software, application techniques, and interface design challenges encountered by today's virtual reality creator. Virtual reality is still in its early stages. It's a lot like the Wild West, and you are a pioneer. The landscape may be fraught with unknowns, even dangers—but we push on, driven by the promise of a better life. Let's take a peek at this new frontier.

**Figure 1-1** shows a screenshot of the now-famous Tuscany VR demo, created by the team at Oculus VR to show off their hardware. Put on the Oculus Rift and launch the demo. You are on the grounds of a Tuscan estate, looking at a beautiful villa. Clouds drift lazily across the sky. You hear birds chirping, and the sound of waves lapping gently against a shore.

You move through the scene, video game-style, using the W, A, S, and D keys on your keyboard (known to gamers as the “WASD keys”). If you play a lot of PC games, this is nothing new. But now, turn your head: looking up, down, and behind, you can see the entire estate. You are there, immersed in a virtual world that completely surrounds you. Walk forward, into the villa, and take a look around. Walk out, up to the edge of the property, and see the lake below. For a few moments at least, you forget that you are not actually in this other place. You’re *present*.

This feeling of total immersion—of being somewhere else, experiencing something else entirely—is what we are striving for with virtual reality. And this is where our journey begins.



*Figure 1-1. Tuscany VR demo by the Oculus VR team*

## What Is Virtual Reality?

Reality is merely an illusion, albeit a very persistent one.

—Albert Einstein

Virtual reality has one goal: to convince you that you are somewhere else. It does this by tricking the human brain—in particular, the visual cortex and parts of the brain that perceive motion. A variety of technologies conspire to create this illusion, including:

### *Stereoscopic displays*

Also known as *3D displays* or *head-mounted displays* (HMDs), these displays use a combination of multiple images, realistic optical distortion, and special lenses to produce a stereo image that our eyes interpret as having three-dimensional depth.

### *Motion tracking hardware*

Gyrosopes, accelerometers, and other low-cost components are used in virtual reality hardware to sense when our bodies move and our heads turn, so that the application can update our view into the 3D scene.

### *Input devices*

Virtual reality is creating the need for new types of input devices beyond the keyboard and mouse, including game controllers and hand- and body-tracking sensors that can recognize motion and gestures.

### *Desktop and mobile platforms*

This includes the computer hardware, operating systems, software to interface to the devices, frameworks and engines that run applications, and software tools for building them.

Without all four of these components, it is hard to achieve a fully immersive virtual reality experience. We will dive further into the details throughout the book; for now, let's take a quick look at each.

## **Stereoscopic Displays**

The main ingredient in virtual reality is a persistent 3D visual representation of the experience that conveys a sense of depth. To create this depth, virtual reality hardware systems employ a 3D display, also known as a stereoscopic display or head-mounted display.

For years, one of the biggest impediments to consumer-grade virtual reality was an affordable stereoscopic display that is **light and comfortable** enough to be worn for an extended period. This situation changed dramatically when the team from Oculus VR created the Oculus Rift. First introduced in 2012, the Rift was a breakthrough in VR hardware, featuring a **stereoscopic display and a head-tracking sensor** built into a lightweight headset that could be purchased as a development kit for a few hundred dollars. While the original development kit, known as the DK1, was fairly low-resolution, it was enough to get the entire industry excited and unleash a storm of VR development. Newer Rift development kit versions, such as the DK2 depicted in **Figure 1-2**, feature higher display resolution, position as well as orientation tracking, and better performance.



Figure 1-2. The Oculus Rift head-mounted display, Development Kit 2

So what does the Oculus Rift actually do? To create the illusion of depth, we need to generate a separate image for each eye, one slightly offset from the other, to simulate *parallax*—the visual phenomenon where our brains perceive depth based on the difference in the apparent position of objects (due to our eyes being slightly apart from each other). To create a really good illusion, we also want to distort the image to better emulate the spherical shape of the eye, using a technique known as *barrel distortion*. The Oculus Rift does both.

A more accurate screenshot of the Tuscany VR demo, a capture of the entire display that renders this scene in virtual reality using these two techniques, would look like [Figure 1-3](#). This shows the rendering as it appears on a computer screen connected to the Oculus Rift's head-mounted display.

From a software point of view, an Oculus Rift application's job is to render an image like the one in [Figure 1-3](#) a minimum of 60 and ideally 120 times per second, to avoid any perceived lag, or *latency*, that might break the illusion or, worse, lead to the nausea that is often associated with poorly performing VR. Exactly how we do this in our applications is the subject of another chapter.

Note that the Oculus Rift is not the only game in town. As we will see in the next chapter, there are several head-mounted displays to choose from. Some of these work only with desktop computers, others are just for smartphones, and still others can only be used with game consoles. The HMDs come in a variety of styles and a range of prices. But as consumer-ready virtual reality displays go, Oculus is the first and—as of this writing—still the best.



Figure 1-3. Tuscany VR demo rendered in stereo for the Oculus Rift

## Motion Tracking Hardware

The second essential trick for making the brain believe it is in another place is to track movements of the head and update the rendered scene in real time. This mimics what happens when we look around in the real world.

One of the innovations in the Oculus Rift is its **rapid head motion tracking** using a high-speed *inertial measurement unit* (IMU). Head-tracking IMUs combine gyroscope, accelerometer, and/or magnetometer hardware, similar to that found in smartphones today, **to precisely measure changes in rotation**. The VR hardware systems covered in the next chapter employ a variety of IMU configurations.

Motion tracking of the head is as important as **quality stereo rendering**, if not more so. Our perceptual systems are very sensitive to motion, and like lag in stereo rendering, high latency in head tracking can break the feeling of immersion and/or cause nausea. Virtual reality IMU hardware must track head movement as rapidly as possible, and the software must keep up. When stereo rendering and head motion tracking are properly combined, and updated with enough frequency, we can achieve a true feeling of being immersed in the experience.

## Input Devices

To create a convincing feeling of immersion, head-mounted displays completely enclose the user's eyes, cutting them off from seeing the outside world. This makes for an interesting situation with respect to input: users have to "fly blind"; they cannot see their mouse or keyboard when using the VR display. To address this, VR is mak-

ing use of alternate types of input devices, including game controllers, hand-tracking motion sensors, and wireless hand and body trackers.

There is currently no one standard way to interact, no “mouse of virtual reality,” if you will. VR is driving a lot of experimentation and innovation with respect to input devices, and we are only at the beginning of that process. Over the next few years, we may see a complete transformation of how we interact with our computers, thanks to VR.

## Computing Platforms

Many VR applications will run on a majority of existing computers and mobile phones. A relatively modern desktop or high-powered laptop can do the trick with Oculus Rift; smartphones can also offer a good VR experience, provided they have enough CPU and graphics power. For most of us, this means that our existing computers and devices can be turned into virtual reality boxes simply by adding a few peripherals. But for those seeking super high production value experiences, the latest desktop PC with a best-of-breed graphics processor and the fastest available CPU might be the top item on the wish list for next Christmas.

As VR matures and gains in popularity, we may also begin to see dedicated computers, phones, and consoles—dedicated, that is, to the sole purpose of enabling amazing virtual reality.

Software to create VR applications comes in several flavors: native software development kits, game engines and frameworks, and even the latest versions of modern web browsers. VR video is another area of exploration. Let’s take a closer look at each of these development options.

### Native software development kits (SDKs)

These are the device drivers and software libraries used in conjunction with the computer’s host operating system. On Windows they would be Win32 libraries used in C++ applications; on Android, they would be Java libraries; and so on. You can build a native application simply using the SDKs, and “roll your own” with respect to everything else (such as 3D graphics and game behaviors). But most developers use engines or frameworks.

### Game engines and frameworks

Unless you are a game engine developer, you probably won’t want to deal with the native SDKs directly; you will more than likely turn to a game engine such as Unity3D (described in detail in [Chapter 3](#)). Libraries like Unity3D, also known as *middleware*, take care of the low-level details of 3D rendering, physics, game behaviors, and interfacing to devices. Most VR developers today build their apps using game middleware like Unity3D.

Many middleware engines have strong cross-platform support, allowing you to write your code once (most of it, at least) and target multiple platforms, including desktop and mobile. They also usually come with a powerful set of tools known as *level editors* or *integrated development environments* (IDEs).

## Web browsers

Much in the way that HTML5 added mobile features over a short number of years, nearly achieving parity with native mobile capability, browser makers are fast following the development of virtual reality. In the case of VR, the adoption of features into web browsers is looking like it will only take one or two years, not four to five.

The upshot of this is twofold: first, it means that we can use web technologies like HTML5, WebGL, and JavaScript to create our applications, making it potentially faster to code and more cross-platform; second, it affords us access to all of the existing infrastructure the Web has to offer, such as hyperlinking between VR experiences, hosting content in the cloud, developing multiuser shared experiences, and integrating web data directly into our virtual reality applications.

## Video players

Stereoscopic video represents a whole class of virtual reality technology by itself. Unlike with game engines, where the application's graphics are completely synthetic, based on handcrafted 3D models, animations, backgrounds, and so on, stereo video is captured from the real world. This makes for truly realistic and often stunning experiences: imagine taking a virtual helicopter tour of the Grand Canyon in VR, being able to look around as you fly over the canyon. Video is not fully interactive in the way that a 3D virtual environment can be though, so the use of this media type has some limitations.

Stereo video recording requires multiple cameras. The minimum is two, but if we want the video to be *panoramic*—that is, capture a 360-degree view of an entire scene for use in virtual reality—then we need even more cameras. Pioneers in the field, such as California-based Jaunt VR, are experimenting with setups that use dozens of cameras to make the first VR feature-length films.

The capture and production of VR video is a nascent field. Several companies and research projects are devoting their efforts to it. There are also a variety of VR video players in development. Some players work only in native environments, while others are only for mobile or the Web, but some developers are creating full cross-platform players. One of the biggest issues in this young endeavor is that there are as yet no standard formats for storing and playing back of the videos, so if you want to produce video content, you may have to choose a single vendor for the hardware, the production tools, and the playback software.

# Virtual Reality Applications

Even though consumer virtual reality is just a few years old, we are already seeing a staggering range of applications. To say that VR has captured people's imaginations would be an understatement. Developers are trying to build virtually *everything* in VR, understandably with mixed success so far.

While it is way too early to pick winners and identify "killer apps" for VR, there are several domains that show great promise, including:

## *Video games*

This is an obvious candidate, the one that most people immediately imagine when you tell them about virtual reality. The potential for deep immersion, higher production value, and stickier engagement has developers, console makers, and peripheral manufacturers salivating in anticipation. It is fair to assume that the majority of skilled, independent, large development shops creating VR are currently doing so for games.

## *Virtual worlds*

Social, user-generated persistent virtual worlds could be a powerful combination with virtual reality immersion. Companies like High Fidelity, created by *Second Life* founder Philip Rosedale, and AltSpace VR, a new San Francisco Bay Area startup, are leading this charge.

## *Education*

For years, 3D visualization has been a great tool for interactive learning; VR immersion could make learning even more approachable and effective.

## *Productivity*

Some researchers and small companies are exploring using VR as a replacement for the decades-old desktop computer metaphor. Imagine a virtual reality, 360-degree workspace that holds your personal information, contacts, work projects, etc.

## *Tourism*

Stereoscopic 360-degree panoramas in VR are really compelling. They represent a simple, effective way to convey the experience of being elsewhere without having to get on a plane.

## *Architecture and real estate*

Architecture and real estate firms are already experimenting with virtual reality, using both video and interactive graphics. Video can be great for showing existing properties; interactive graphics work well for visualization of buildings and complexes in the planning stages (i.e., using 3D CAD models).

### *Live events*

VR video is promising to be quite popular for concerts, news reporting, and other live events. Musicians Paul McCartney and Jack White are among several rock stars who have already broadcast virtual reality versions of their live shows.

### *Web browsing*

Mozilla is leading the charge in experimenting with adding VR support to its browser, and Google is not far behind in creating the same features for Chrome. Beyond building the technology plumbing, the research team at Mozilla is exploring visual and interface designs for how to navigate a universe of information in virtual reality.

### *Enterprise applications*

There are countless potential VR applications for the enterprise, including simulation and training for military use, medical diagnostics and training, and engineering and design.

This list is just a sampling. We have no way of knowing which applications will be most successful in a new medium like virtual reality, or what will become popular. We also can't predict what other ideas will spring from the minds of clever folks—but if the last few decades of technology have taught us anything, it's that whatever people come up with, it will probably be something we never could have imagined before.

## **Chapter Summary**

Consumer virtual reality is upon us. As we have seen in this chapter, VR brings together several technologies, including 3D stereoscopic displays, motion tracking hardware, new input devices, computers, and mobile phones. The key innovations that enable virtual reality are stereoscopic rendering and motion tracking. When these two are properly combined, we feel immersed, or *present*, and the illusion of VR is compelling enough to transport us to another place.

There are a variety of development options for virtual reality: native SDKs that access raw platform features; game engines that give us more power and cross-platform reach; web browsers, for creating shared, connected virtual reality apps; and video players and tools for creating stereoscopic panoramic videos based on the real world.

Though the technology is still immature, virtual reality is already being used in a number of applications, from games to education, from real estate to rock concerts. Developers are also creating productivity and enterprise applications, looking to the power of VR immersion to enhance business value.

In the next chapter we will explore the wide variety of virtual reality hardware systems in use today. Let's get to it.