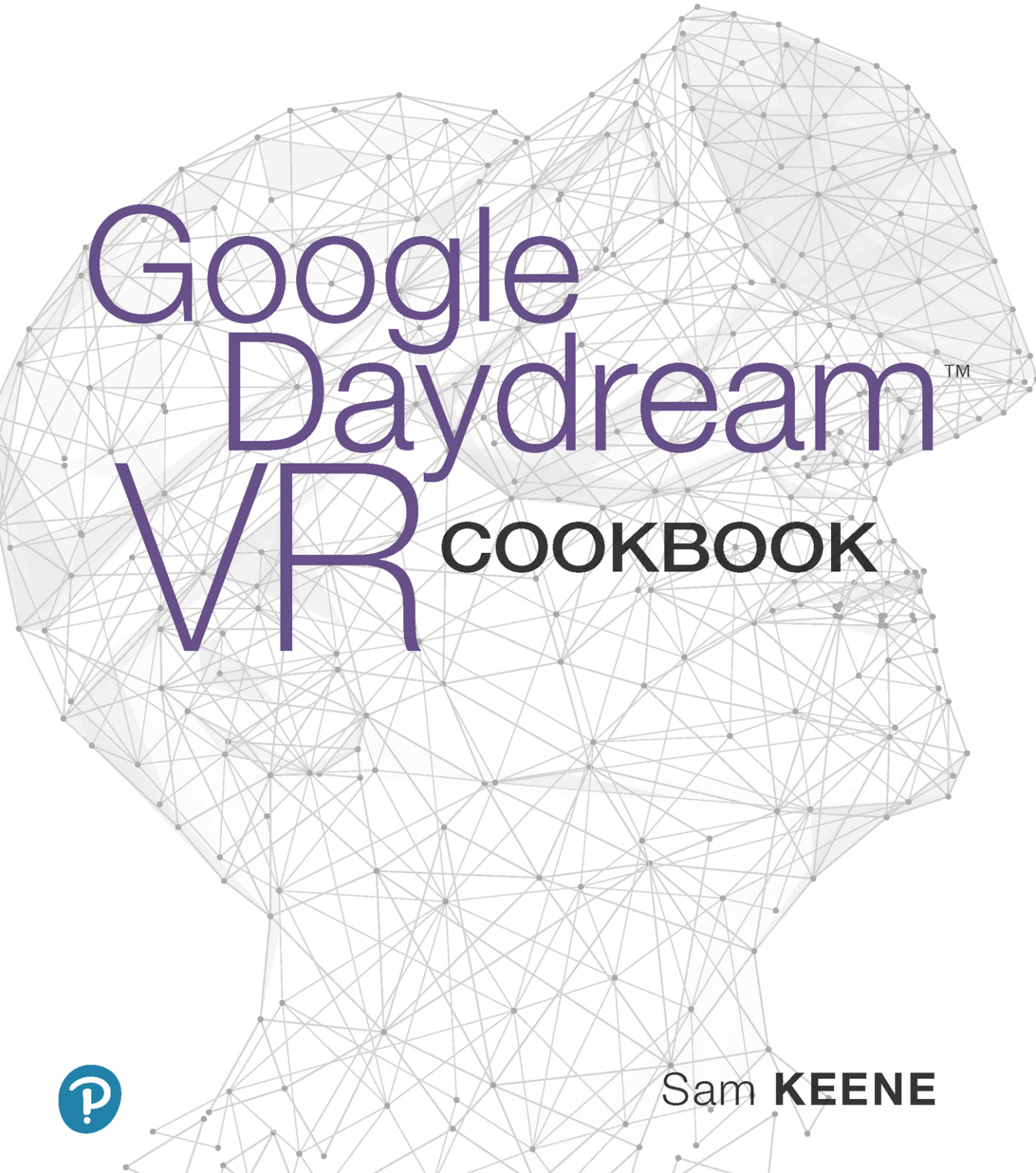
BUILDING GAMES AND APPS WITH
Google Daydream AND UNITY



Google Daydream™ VR COOKBOOK

Sam KEENE

Google Daydream™ VR Cookbook

Designing User Interfaces in VR

Designing great user interfaces (or UI) for VR is a deceptively challenging task. Unlike other digital mediums, such as mobile or desktop, where creative possibilities are constrained by screen sizes and flat surfaces, the potential permutations of interfaces in virtual reality are limitless.

As a VR creator, the process of building UI requires a hybrid skillset encompassing design and engineering. Underpinning these two disciplines are other considerations spanning user experience (UX), art direction, psychology, ergonomics, and software engineering. Each of these areas offers a complex challenge in their own right, but when combined in the medium of VR, they become an ambitious undertaking.

Additionally, people's expectations for product excellence in VR have been set very high. Being experts in everyday "meat space" reality (also known as everyday life), we expect a higher quality in VR because of the extra possibilities it affords. With AAA game play, high-end graphics, and special effects, people anticipate an enhanced experience.

Simply creating a functional, usable experience is then not enough. It also needs to be aesthetically beautiful and innovative, empowering the user to do more than the options they have in real life by giving them playful agency over the world. Many VR games successfully walk this line, but it is really not an easy one to tightrope along.

With that said, the most important thing is to get the basics right. This chapter focuses on creating spatial UI rendered in world space using Unity's built-in UI system. World space UI in Unity is simply UI rendered as 2D panels, placed in the 3D environment.

By the end of this chapter, you will have built a template for a game menu that can asynchronously load in game levels, as well as a complex scrolling panel system for an app that lays out content similar to the UI in the Daydream Home world (see Figure 4.1).

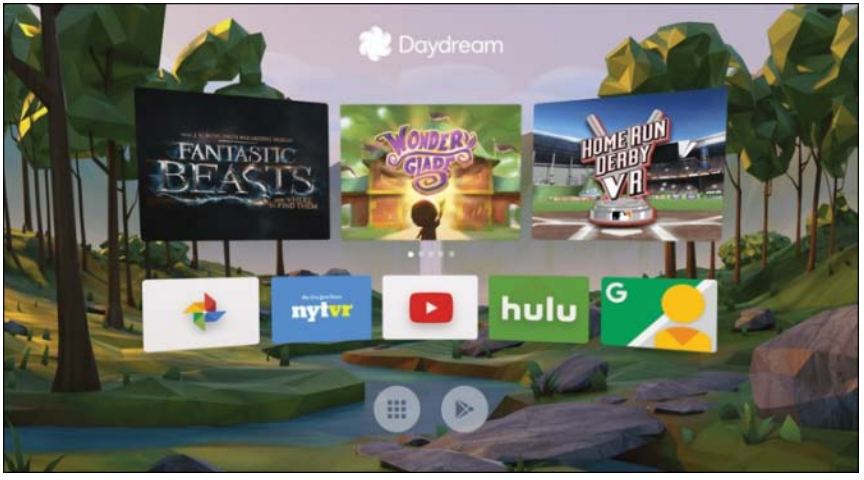


Figure 4.1 The Daydream Home world. (© Google LLC)

Even with the multitude of possibilities in VR, it turns out that the easiest way to consume basic information is the same way we do in the real world: on flat rectangular screens and often in text format.

This section discusses various principles of design worth considering when creating UI for these flat rectangles. The design system and workflow developed by Google's Daydream design team is then presented as a way to apply these design principles in practice.

Design VR, Use VR

It sounds obvious, but the best way to design for VR is to use VR. This means not only systematically testing your work in the VR environment, but researching and exploring what other people are doing in VR; in other words, immersing yourself in it. You cannot create well-designed UI if you are not using VR regularly.

User Experience Design in VR

The principles of human-centered design hold true in virtual reality just as they do in any other design-related medium.

The goals and needs of the user are paramount to the experience, and as such they should drive the design process across all stages of product development. Good design in VR is a constantly reiterative process of evaluation, creation, and testing: as is all good design, essentially.

The overarching principle of human-centered design1 (or HCD) is of working out the goals and requirements of the user by placing them at the center of all design thinking. Through this methodology the system is analyzed, pain points for the user are identified, and an attempt is made to transform difficult tasks into simple ones.

The usability of a VR system can be tested against classic HCD principles in the same way as they are in any other traditional design practice.

When iterating on interface design, VR designers should be looking for ways to simplify complex tasks by

- Limiting choices to, where possible, the correct ones.
- Applying clear mapping between the relationship of a UI element, and what it actually does—its function or outcome.
- Making sure enough information is available in the world for users to build a clear mental model in their heads of how the system works.

1. *The Design of Everyday Things,* Donald A. Norman.

- Designing for error. People make mistakes, particularly in VR, and the system should not punish human error, lock people out, or become stuck in a state that is difficult to exit out of.
- Falling back on convention. When the design solution is unclear, use an existing or industry standard. If an accepted way of doing something already exists, then do not reinvent the wheel.

Ergonomics

The particular ergonomic factor of users' sitting position and hand position plays a significant role in how they will interact with UI in Daydream VR. Most importantly for us, the primary location of most users when using Daydream VR is lounging on a couch or some similar furniture.

Different viewing positions and postures affect the users' gaze and focus on the screen. Daydream users tend to tilt their head down slightly by 10 to 15 degrees, while their eyes look up, moving the center point of focus to around 6 degrees below the horizon (see Figure 4.2).

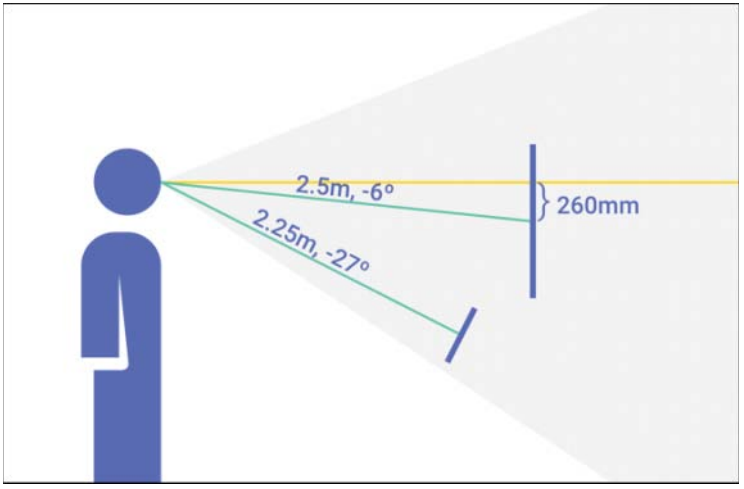


Figure 4.2 A user's comfortable front viewing area as viewed from the user's side on.

Although they are seemingly minor, being aware of these factors is a crucial step to improving usability and should be considered with a high degree of importance when deciding on placement of UI.

Material Design

Material Design is Google's design language created to visually unify all of Google's various products and services.

It uses the concept of material planes as a visual metaphor that builds a theory for rationalizing interfaces in space and a system for how they should move.

Although Material Design was not made specifically for VR, Daydream's design team has done some great work in applying Material Design principles to Daydream's UI.

Find out more about Google's Material Design system by visiting https://material.io/.

Daydream's Design System: dmm

A smart way to design user interfaces for VR is to base them around the legibility of text.

However, the further UI moves away from the user in 3D space, the less coherent it becomes. How can a designer create UI in a 2D application that will be clear and readable in VR at any distance?

The Daydream UX team has developed a system and workflow that solves this problem and makes working with UI consistent across different screens, 2D and 3D, and at any distance. At its core, it uses a system of angular units.

Conceptually, at an atomic level, virtual screens and real screens share the same properties. They have content that needs to be displayed, and they each have an intended viewing distance and size.

In the real world, when you look at a billboard across the street and then hold your phone up next to it, the text on each of those screens will be legible and appear to be at a similar size. The same principle applies to UI in VR. As things move further away, they must be scaled up to stay legible. Text in the foreground and background look the same if they have the same angular scale. This phenomenon of angular scale is the core concept behind Daydream's design system.

The Daydream team's system for designing comfortably legible UI using angular scale is called *distance-independent millimeter* (dmm, pronounced "dim"). A dmm is described as one pixel (in your 2D design application) being equal to one millimeter at one meter distance away in VR. The scale then increases based on the distance. So, at 2 meters distance it doubles to 2mm, at 3 meters it triples to 3mm, and so on (see Figure 4.3).

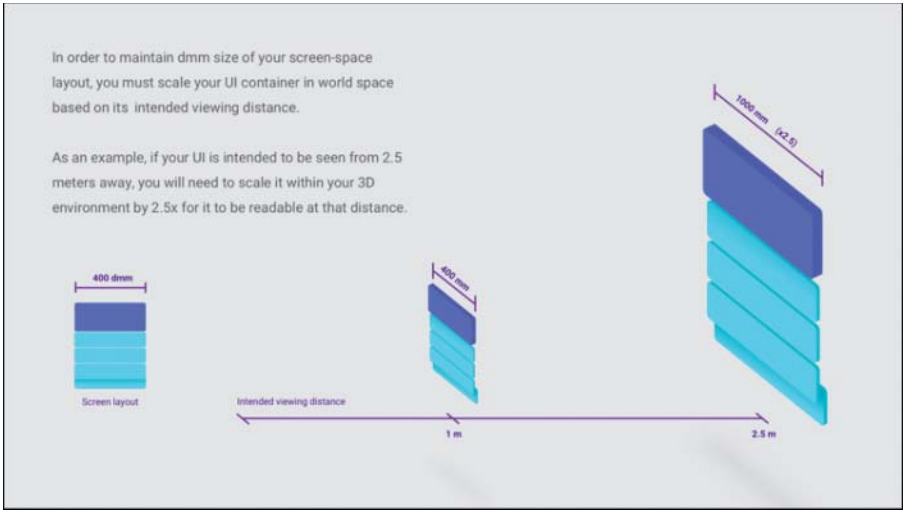


Figure 4.3 With dmms, UI needs to be scaled by a factor of its distance.

This elegant solution means you simply scale the UI by a factor of its distance from the user, making it legible at any distance. As a designer, you can have confidence your pixel precision mocks will look the same in VR as they do in Sketch.

Sketch

Sketch is a design application that has become popular with digital designers in recent years. Because of its simplicity for rapidly mocking up mobile app interfaces, Sketch has quickly become the industry standard. It turns out that Sketch is also great for designing UI in VR.

You can download a free trial of Sketch from https://www.sketchapp.com/. I recommend you install it along with the Daydream Sketch UI sticker sheet (https://developers.google.com/vr/design/sticker-sheet). Sketch is currently only available on Mac.

To help designers easily apply the concept of dmms to their VR UI in Unity, the Daydream design team has open-sourced a Sketch template that is based in dmms (see Figure 4.4). It has standard minimum text sizes and various Material Design and UI components included. They all conform to the system of dmms, so your interfaces will transfer easily from a 2D application into Unity's 3D space, and they will look correct at any distance.

If using dmms still sounds complicated, do not worry. The first recipe in this chapter steps you through the process of creating a template using dmms to handle UI in Unity, with textures exported from the sticker sheet. Go ahead and download the Sketch sticker sheet from the Daydream VR developer site at https://developers.google.com/vr/design/sticker-sheet.

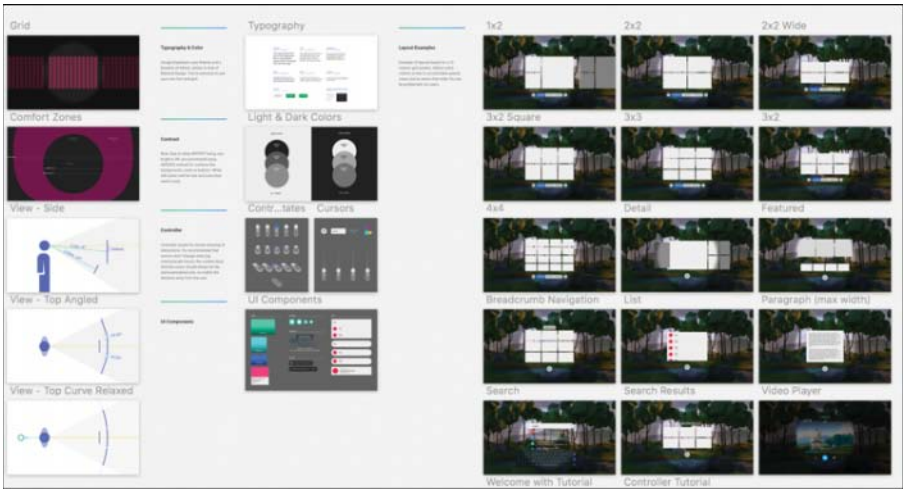


Figure 4.4 The Daydream UI sticker sheet. (© Google LLC)

UI, Unity, and the Google VR SDK

Moving on from UX and design concepts, let's head to the practicalities of actually building the UI in Unity. As in the previous chapters, the methods described here use a combination of Unity's built-in components and those provided by the Google VR SDK.

Unity has its own robust system for laying out complex UI, conveniently called the UI System. It comes with standard components for visual elements, layouts, interaction, buttons, and other useful controls.

In Daydream, these components are made interactive using the custom Event System provided by the Gvr SDK, the `GvrPointerGraphicsRaycaster`. The Event System is the same one outlined in the previous chapter, with a minor addition of the custom Graphics raycaster, as opposed to the Physics raycaster.