David Herman
senior software engineer | tech lead | mentor

bitspittle@gmail.com
Mountain View, CA 94041

I'm a programmer with 20 years of experience, the first 10 in the game industry and the rest at Google. As a tech lead, I have a track record of delivering software features on time. Besides programming itself, I enjoy navigating the dynamics of teams: enabling collaboration and autonomy in others, generating psychological safety, and providing mentorship.

Employment

Google - Android

2014-2021

Worked on Android Studio, an IDE for Android development. Historically Java code with lots of new code written in Kotlin.

- Part of a team of eleven engineers who designed and built a collection of Android profiler tools from scratch, consisting of CPU, memory, network, and energy monitors. This included work on a C++ daemon that ran on an Android device, Java code (also on device) that instrumented the user's app at runtime via bytecode instrumentation, and Java client code on the host for data polling and UI. Lead a team of two other engineers on the network and energy profilers.

- Led an effort of two other engineers for the IDE integration of a feature called Data Binding, picking up code that had mostly been abandoned two years earlier when an engineer left the team. I introduced a process so our team could discover and prioritize which issues to tackle. While the original code was Java, I got support to write all new code in Kotlin, one of the first subteams to do so. After a year, the codebase was stable, thoroughly documented, well tested, and much easier to maintain. User surveys collected a year apart showed that we significantly improved user satisfaction.

- Led a team of two other engineers implementing a framework called App Inspection, which lets you write a library inspector that runs against a user's app and communicates otherwise opaque state back to Studio, where it is rendered by UIs. This API was written in Kotlin and uses coroutines for all async behavior. The framework has allowed us to add database inspection, a feature long requested by users, and more support is landing soon for visualizing tasks that run in the background.

Google - YouTube

2012-2014

- Worked on a C# app for the Xbox 360 using Silverlight. Our team helped YouTube grow the (at the time) nascent living room space, with our own product gathering 3 million watch hours daily by the end of two years.

- Refactored the codebase to allow seamless migration over to a significantly different upstream API.

Sony Playstation - MLB: The Show

2003-2012

- Built a UI Editing tool from the ground up in C#, working with 6 artists who used it to craft over 200 UI screens per target platform per year (at one point supporting three platforms simultaneously - PSP, PS2, and PS3). All UI data, including animations and textures, exported to a lean couple dozen kilobytes, vs. an industry standard tool used by a different team in the same building which generated similar data files with sizes in the 100s of megabytes range.

Sony Online Entertainment - EverQuest

2001-2003

- Maintained a C++ login server that load balanced a playerbase of 400,000 monthly active users, serving as a chat area for talking to customer support or just killing time with other players if the game servers were down.

- Helped migrate all game UI from overly coupled, hardcoded C++ code to a flexible XML system.

Github Samples

I regularly work on hobby projects during my time off, as a way to explore ideas or technologies that are unrelated to my day job. Most projects are unfinished, as work inevitably comes back and takes priority, but I'll include a selection of them here so you can see what my code looks like even when there isn't anyone looking over my shoulder.

https://github.com/adt-dherman/DoomFire.kt - A coworker wrote an article about how the title screen in the 1993 game *Doom* employed a simple but effective fire animation. This project implements that effect in Kotlin using TornadoFX for the UI.

https://github.com/bitspittle/lively - A Kotlin library that allows you to set up relationships between variables in a safe and intuitive manner, where observers automatically update when dependent values change.

https://github.com/bitspittle/truthish - A library inspired by Google Truth with some edge cases improved by leveraging features unique to the Kotlin language.

https://github.com/bitspittle/kross2d - Inspired by ggez (a Rust 2D game library), this was an experiment to write a simple 2D game engine in Kotlin Multiplatform, targeting desktop and web.

Other Materials

https://medium.com/androiddevelopers/learn-kotlin-through-unit-tests-914106d2d8c5 - A medium post from a document I originally wrote for my own team on how to get started with Kotlin.

https://www.youtube.com/watch?v=LGVbpobV-Yg - A talk I co-prepared, teaching advanced profiler tips.

https://www.youtube.com/watch?v=rjlhSDhFwzM - A talk I co-prepared, presenting our team's favorite Android Studio debugging tips.

Languages

Very comfortable with: Java, Kotlin
Work experience with: C/C++, C#, Python
Personal experience with: Coffeescript, Dart, Javascript, Typescript, GDScript, Lisp, Rust

Education

1997-2001 Harvey Mudd College, B.S. Computer Science