
unity

Who am I

Matti Frimodig https://www.linkedin.com/in/mattifrimodig/

Member of the Wallet Team

Professional Software Developer since 2006

Using Kotlin since 2016 / v. 1.0

Joined Unity June 2018

Kotlin Basics

Generative Art – Made with Unity

unity

What is Kotlin

Kotlin has been created as a drop-in replacement for Java. It strives for a better balance between functional and object-oriented programming styles.

Benefits similar to other non-Java jvm languages:

- Less boilerplate code
- More concise syntax
- Removing(/hiding) internal inconsistencies that Java has accumulated
- Maintain Java interoperability

What is Kotlin - Type Inference & Null Safety

```kotlin
import kotlin.math.sqrt

fun main() {

    val map = mapOf("one" to 1, "two" to 2, "three" to 3) // Map<String, Int>

    val map2 = map + ("four" to 4.0) // Map<String, Any>

    val int = map["one"] // Int?

    val int2 = map2["two"] // Any?

    val sum = if (int != null && int2 is Int) {
        int + int2
    } else null

    val squared = sqrt(sum?.toDouble() ?: 1.0)
}
```

Why Kotlin

Kotlin as a backend development language in financial software

“Indeed, the ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code. ...[Therefore,] making it easy to read makes it easier to write.”

— Martin C. Fowler, Clean Code

Safety

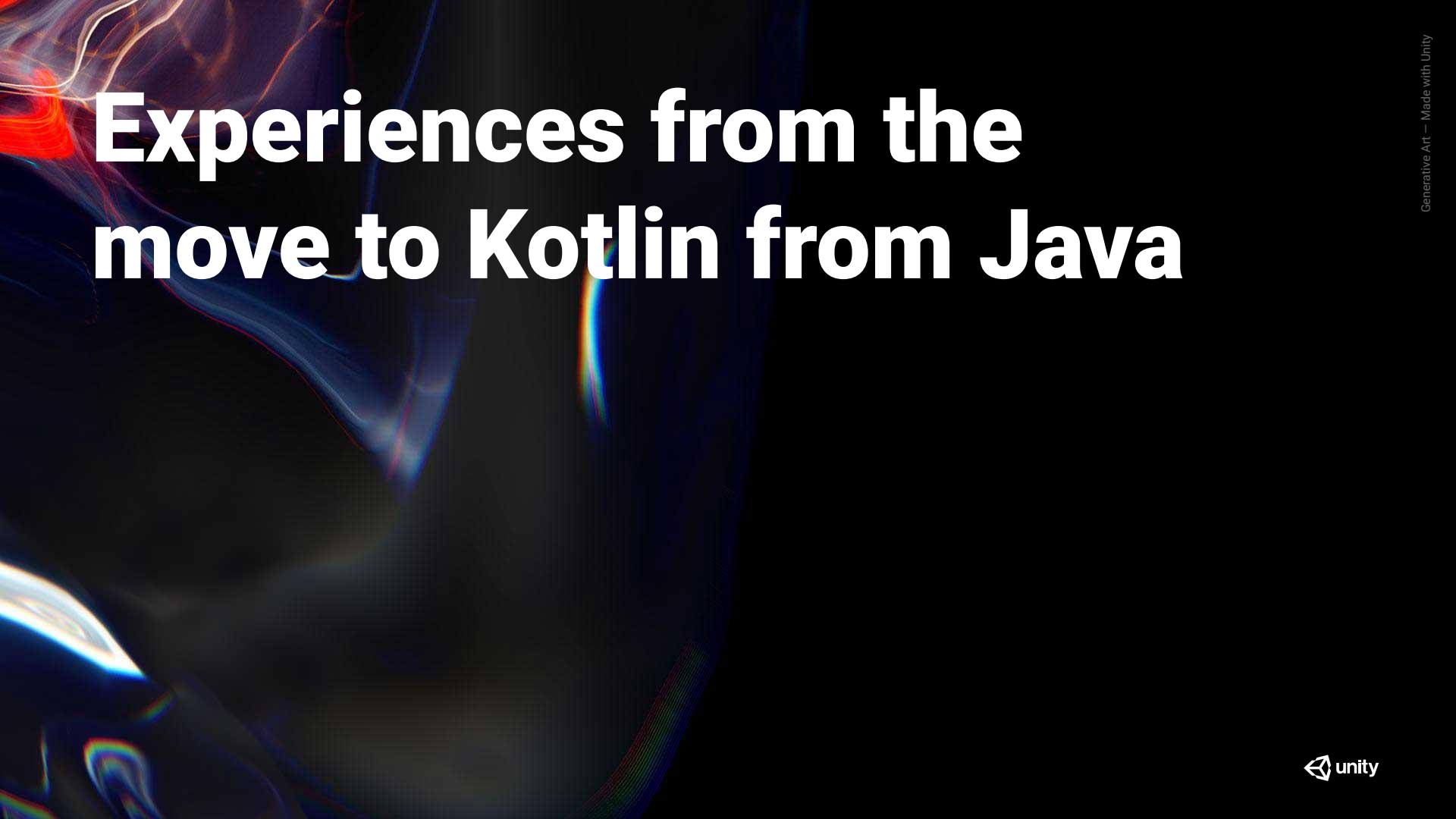
Kotlin helps us to write safer code by

- Adding null safety
  - Makes it harder to accidentally create nullability issues
- Limiting mutability
  - Variables and collections have to be explicitly declared mutable
- Making verifications more explicit
  - Provide a standard way of coding state & parameter verifications making it easier to check that they are in place

Programmer Happiness

- Developers wanted change
- Everyone loves working with cool new things
  - fastest-growing programming language, according to GitHub
- Some things were needlessly complicated in Java
- Good experiences / word-of-mouth
- Striking the right balance
  - Statically typed
  - Designed for readability

Experiences from the move to Kotlin from Java



Generative Art – Made with Unity

unity

Findings

- All the developers in the team feel that the move to Kotlin was beneficial
- The overall team productivity suffered while people were learning
- A bunch of libraries were replaced in the tech stack as a side effect
  - More Kotlin friendly alternatives
  - More lightweight
- Creating a best practice document was helpful
  - Useful overview
  - Architecture decision log

Good and Bad Features

- Good
  - Null safety
  - Things that reduce boilerplate code
    - Data classes
    - Default arguments
    - Type inference
  - When expression
- Bad
  - Platform types returned from Java libraries

Transition pain points

- Unfamiliar syntax
  - Java like, but not like Java
    - Lack of explicitness can detract from readability
    - Minor logic changes like implicit returns
  - Understanding scope functions
    - let, run, with, apply, and also
- Learning to use the core library
  - Many functions that were in different java libraries are now available though the core library, but the IDE isn't very good at driving you to use them
- Uncertain tool support

Kotlin compared to Java - code examples

https://github.com/frimodig/kotlin-samples

Resources

- https://kotlinlang.org/docs/reference/
- Effective Kotlin by Marcin Moskala
  - https://leanpub.com/effectivekotlin
- Kotlin in Action by Jemerov & Isakova
  - https://www.manning.com/books/kotlin-in-action

Thank you.

#unity3d