# Google Summer of Code 2020

# Support for Kotlin Native

## Aditya Rana

E-mail : mail2aditya.rana98@gmail.com
Country : India
Timezone : IST (GMT + 5:30)
GitHub : github/ranaaditya
Linkedin : linkedin/adityarana
Resume: resume/adityarana
Keybase : keybase/ranaaditya

## Project Abstract

I propose to work on following:

1. Migrate the existing **Processing Android Library** to **Kotlin**.
2. **Restructure** the existing **android library** to a **multiplatform library** using **Kotlin Native**.
3. Upgrade the **Groovy based Gradle** build System to **Kotlin Script based (kts) Gradle** build System.
4. Splitting android-specific logic from Java.
5. Make a common **Kotlin interface** used by multiple platforms (android and iOS).
6. **Implementation** on at least one platform (primarily **JVM**).

# Project Description:

## Support for Kotlin Native:

This project proposal is based on the **Support for Kotlin Native** for **Processing Android mode core libraries,**as described in the Processing Foundation's [project list](#) . In summary, the proposed work in this document is divided into 3 stages:

- **Migration** of the Processing Android mode core libraries into their Kotlin Native.
- **Restructuration** of the existing code in order to support the iOS. In short, restructure the existing android mode core libraries into a multiplatform library using Kotlin Native.
- **Implementation** of the restructured multiplatform library on atleast Android JVM and then, on iOS with some stubbed methods. This will play a significant role to permit Processing Android mode to keep up with the changes in the Mobile development (iOS and Android) technology. The Restructuration andImplementation will also be helpful for the beginning of Kotlin Native as a mode, which will permit users to target the iOS market, thus, encouraging the next generation of developers. This exactly aligns with the core principles of the Processing Foundation to close the technological gap and bring programming to the masses.

The complete description of aforementioned 3 stages is as follows:

## Migration:

This will be  the very first stage of my internship. This stage includes the migration of Processing-Android's core android libraries into Kotlin equivalent.

This is a big task to convert all the core  android library code into Kotlin,  and will be carried out with the  [Kotlin clean architecture](#). This migration also

includes the Migration of AR and VR libraries existing inside the processing - android codebase from Java to Kotlin.

**Restructuration:**
This is the second stage of my internship. This stage includes restructuring the existing Processing-android  codebase into a [multiplatform library](#), as described in the project list so that users can also use it for  iOS as well. It includes splitting of Android specific logic in Java and restructuring rest platform independent code into Library with KotlinNative.
It also includes conversion of the Groovy based Gradle build System to Kotlin Script  based build Gradle System.I tried a [multiplatform-app](#) using Kotlin and Kotlin Script for Gradle build.

**Implementation:**
This is the third and final stage of my internship, this stage consists mostly of testing phase of our restructured multiplatform library on at least one platform (primarily on JVM) and  will do it for iOS. This phase also includes writing of **[Integration Tests](#)** files for libraries. After completion of these tasks, I will start writing sample apps for our work to be included in the Kotlin Native mode for ease of the user to understand our work.

---

**DEVELOPMENT PROCESS:**

**Time period for Community bonding** (May 4 - June 1):

- Familiarise myself with the code base of Android mode, core libraries and the existing Library Template.
- Commence discussion on topics listed in the proposal above.
- Finalise deadlines and milestones.
- Interact with the mentor and decide upon the project structure for libraries, and contribute to the code base whenever possible.

- Analyze the updates regarding version changes in Kotlin and their new updates for Kotlin native and kts build tools.
- Identify more issues on the multiplatform using Kotlin, reported by various users on platforms like stackoverflow, github, etc.

**Coding Period** (June 1 -Aug 24):

- **Phase 1** (June 1 - June 29) :
  During this phase, migration of existing code base will take place including the migration of, first, the helper classes, then, Processing-Android core libraries. After completing this, migration of AR and VR libraries existing inside the Processing-Android codebase will take place. After completing this, I will do rigorous testing of the proposed changes in this phase which includes writing of [Unit Test](#) for each migration and real time testing using the Android platform. After testing, writing of Documentation for the proposed work will take place.

- **Phase 2** (June 29 - July 27):
  This phase includes the restructuring of the migrated codebase into a multiplatform library. So, other platform users can also use processing. Splitting the android-specific logic in Java and converting rest into Kotlin Native powered library. This phase also includes writing of build files in Kotlin Script for the previous libraries migration to Kotlin which were written in Groovy based Gradle. After completing this, I will do the testing of the proposed work. After completing this, writing of documentation of the restructured code will be done .

- **Phase 3** (July 27 - Aug 24):
  Third and last phase of the internship program includes the Implementation of this **multiplatform library** on at least one platform (primary will be Android (JVM)) will be done. After this testing, I will continue this implementation on iOS. Then, I will try to make sample apps for iOS using the multiplatform library proposed like the basic apps in the Processing-Android repository for reference. If I will get

buffer time during this phase then I will start converting [Video Library](#) as mentioned in ongoing projects list from Java to Kotlin and rewrite its Groovy based build files with Kotlin Script (kts) based build Gradle files.

---

## Timeline:

### Week 1-2 (June 1 - June 15):

- Identify the codebase that has to be migrated to Kotlin, and start migrating to Kotlin.
- Mostly migrating the helper(data) classes first, so no breaking of code will happen and spend most time in migrating other related stuff.
- Start migrating core android library code this week and migrate the event handling code as well.
- Start writing the [Test File](#) for migrated code.
- Test the migrated code on android platform.

### Week 3-4 (June 15 - June 29):

- Start migrating the OpenGL code into Kotlin.
- Migrate the rest libraries one by one.
- Migrating the AR and VR libraries into Kotlin.
- Start writing the test files for the migrated code.
- Test the migrated code on android platform.
- Write [documentation](#) for the migrated Kotlin code.
- With this, I will submit my work on Migration of Java code into Kotlin.

### First Evaluation (June 29 - July 3):

- Migrated and restructured code will be reviewed by mentors.
- Discuss the changes with the mentor and implement them.
- Make sure that none of the code is breaking and continue the testing of code on the android platform during this period.

- Will do any left work regarding the first two stages.

---

## Week 5-6 (July 3 - July 13 ):

- Restructuring the migrated codebase into a multiplatform library using Kotlin Native.
- Split the android-specific logic from plain Java and make a **common Kotlin interface** used by multiplatforms.
- Restructuring the build files for migrated core android library code from Groovy based Gradle build System to **Kotlin Script (kts) based Gradle System**.

## Week 7-8 (June 13 - July 27):

- Start writing the build files in Kotlin Script for AR and VR libraries that were migrated to Kotlin.
- Includes writing of theIntegration Tests for the build system of both AR and VR libraries in Kotlin Script.
- Discuss this issue [iOS OpenGL](#) with mentors and how it can affect our ongoing work.
- Start writing the test files for the code.

## Second Evaluation (July 27 - July 31):

- Restructuring the library code and conversion of build system code will be reviewed by the mentor.
- Make sure that none of the code is breaking and continue the testing of code on the android platform during this period.
- Will complete any left work of the second stage.

---

## Week 9-10 (July 31 - Aug 10) :

- Start implementing the  restructured multiplatform library for the Android Platform (JVM) first.
- After successful testing of code on JVM, I'll move to iOS implementation. It includes testing of libraries with at least a few stubbed methods of the API implemented in iOS for proof of work.

## Week 11-12 (Aug 10 - Aug 24):

- This is a buffer week in which I will finish any left over work of my internship.
-  If time permits I will continue with conversion of this [video library](#) ongoing project as mentioned in the [project list](#) into Kotlin.
- It involves rewriting build files in Kotlin Script and making it a multiplatform library with one platform specific as of now (primarily JVM).
- If time permits me, I'll continue the implementation of video library in iOS as well.

## Final Evaluation and submission of Project (Aug 24 - Aug 31):

- My final part would be to document all the new changes that I have made to the project so far.
- Submit all my contributions as part of my final evaluation.
- Start preparing GSoC blog as a part of my proposal.

## More about me :

I am **Aditya Rana**, an undergraduate student at **National Institute of Technology , Tiruchirappalli** (**NIT Trichy)**, currently pursuing my second year of B.Tech.
From the very first year at my college,I have been involved in Application Development (both native and hybrid) but it is operating system that I found

myself keenly interested in. This prompted me to try developing my own [OS](#) from scratch.

Apart from my personal interest in application development and operating system, I am a functioning member of [DeltaForce](#) , the premier coding club and **official webteam** of **NIT Trichy**, where I work as a Full-Stack Developer. With Delta,I got innumerous opportunities to develop several android apps. It was heartwarming to see the apps I have developed used by more than 10,000 college students in various online events and competitions during college fests. Exposure to the great learning environment in DeltaForce encouraged me to participate in competitive events and hackathons. A few projects that I have developed can be found on [github](#).

I was first introduced to Processing Android when I started using Android Studio for apps. I used **Processing Sketches** for **Splash Screens and** found it to be very useful in my quest to develop an app for Pragyan, an ISO certified annual technical fest of NIT Trichy. I also made a **3D particle system** for the main screen of the same app. I also tried my hands on developing a 2D ping pong game using processing with help of [ tek gadg tutorial](#) .

Since learning never stops,even after getting well versed with Java, Kotlin, Golang, python, c, c++,I have spent a lot of time getting familiarized with the existing Processing codebase. Amidst all these opportunities, I found myself climbing up the steep learning curve and am always willing to embrace further opportunities coming across my way.

## Following are my contributions to OpenSource:

### Processing Foundation:

- Opened issue [#6011](#) related to a build problem that I faced during the setup of the processing.

### Delta Winter of code:

- **Mentored** a group of students from various colleges of India for a **BillSharing** [App](App) under the **Delta Winter of Code'20** [(DWoC)](DWoC) program.