droidcon academy

Reducing App's APK Size in Android

Musab Nasreldin

@musabnasreldin

in/musabagab

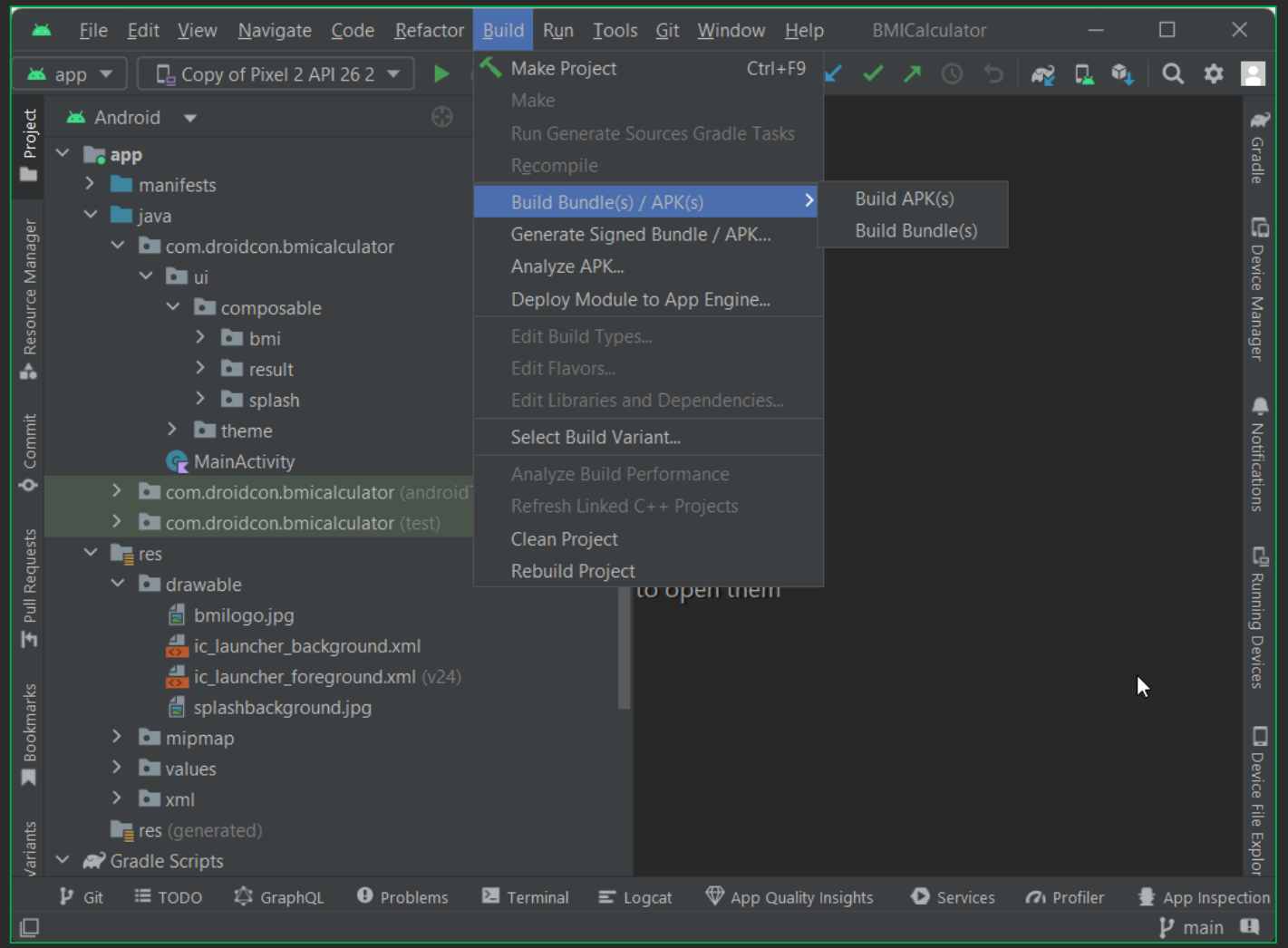
Importance of Optimizing APK Size

- Faster Downloads and Updates
- Reduced Storage Space
- Improved Installation Rates

Advantages of Android App Bundles

- Dynamic Delivery of Features
- Smaller App Size
- On-Demand Loading

Building Android App Bundles



Finding Unused Resources using Lint

- What is Lint?
- Where can it be used?
- What does Lint do?
- Lint Report Result

Vector vs. Other Image Formats

- Scalability and Resolution Independence
- File Size and Loading Speed
- Complexity and Compatibility

WebP Format

- Smaller File Size
- Lossless and Lossy Compression Options
- Alpha Transparency Support
- You can convert other formats to WebP via Android Studio

Code and Resource Shrinking : R8 and Proguard

- Unused Code and Resources Removal
- Obfuscation and Protection

Code and Resource Shrinking : R8 and Proguard

```
android {
    buildTypes {
        debug {
            minifyEnabled true
            shrinkResources true
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),'proguard-rules.pro'
        }
    }
}
```

Challenge Discussion : R8 vs Proguard

- R8:
  - Default code shrinker and obfuscator starting from AGP 3.4
  - Faster build times compared to Proguard
  - Provides more advanced optimization techniques

- Proguard:
  - Traditional code shrinker and obfuscator used in Android development.
  - Longer history and a larger user base compared to R8.
  - Offers extensive configuration options for code obfuscation and optimization

Summary

- Importance of optimizing APK size
- Building Android App Bundles
- Identifying and removing unused resources using Lint
- Comparing Vector and other image formats
- Creating WebP images
- Code and resource shrinking using R8 and Proguard