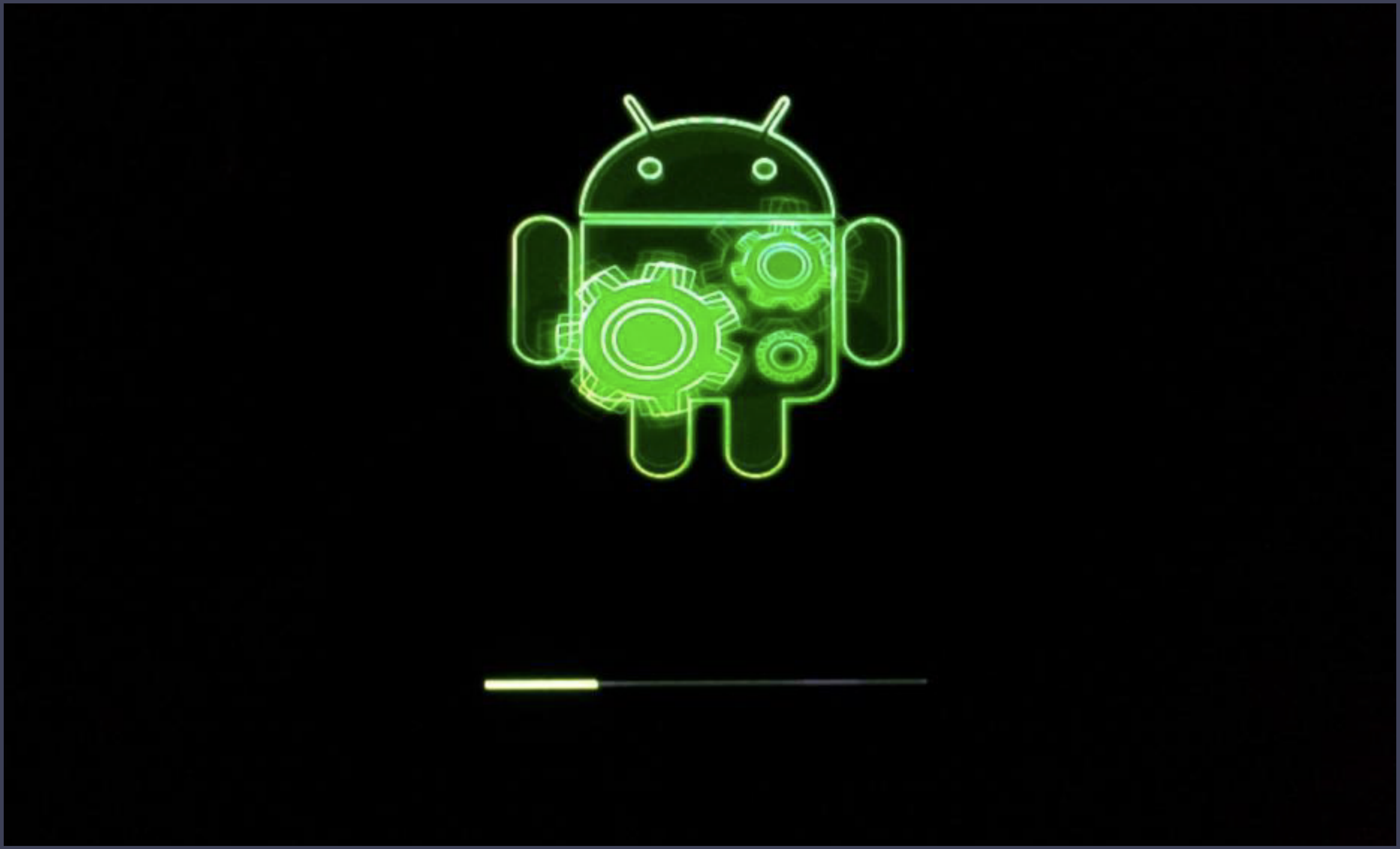
Version Management

@KeithYokoma - Drivemode, Inc.

potatotips #11

KeithYokoma

Keishin Yokomaku
Drivemode, Inc.
Android Engineer
GitHub: https://github.com/KeithYokoma
e-Book: http://amzn.to/1mZNydv



When your application is updated…

You may…

- Change database scheme
- Write new preferences
- Migrate to new data models
- Trigger data synchronization
- and so on…

It could happen!

```
int current = // get current version code from PackageManager
int previous = // read previous version from Preference
if (current <= previous) { return; }

if (previous < 1) {} // do when the app is updated from 0
if (previous < 2) {} // do when the app is updated from 0 or 1
if (previous < 3) {} // do when the app is updated from 0 ~ 2
if (previous < 4) {}
if (previous < 5) {}
if (previous < 6) {}
```

It could happen!

```
int current = // get current version code from PackageManager
int previous = // read previous version from Preference
if (current <= previous) { return; }

if (previous < 1) {} // do when the app is updated from 0
if (previous < 2) {} // do when the app is updated from 0 or 1
if (previous < 3) {} // do when the app is updated from 0 ~ 2
if (previous < 4) {}
if (previous < 5) {}
if (previous < 6) {}
```

Too many “if” statement!

```
int current = // get current version code from PackageManager
int previous = // read previous version from Preference
if (current <= previous) { return; }

if (previous < 1) {} // do when the app is updated from 0
if (previous < 2) {} // do when the app is updated from 0 or 1
if (previous < 3) {} // do when the app is updated from 0 ~ 2
if (previous < 4) {}
if (previous < 5) {}
if (previous < 6) {}
```

Just fit for you

https://github.com/KeithYokoma/Fit

Fit

- Annotation based
- Automatically call your procedure
- Reusable method declaration
- No more dull “if” statement

Version Module class

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Version Module class

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Version Module class

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Application class

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fit.initialize(this, new MyModule());
        Fit.getInstance().execute();
    }
}
```

Application class

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fit.initialize(this, new MyModule());
        Fit.getInstance().execute();
    }
}
```

Application class

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Fit.initialize(this, new MyModule());
        Fit.getInstance().execute();
    }
}
```

That’s it!

Behaviour
Version 1 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 1 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 1 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 1 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 1 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 3 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour
Version 3 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Behaviour Version 3 to 4

```
public class MyModule implements VersionModule {
    // foo() is called when the app is updated to
    // version code = 1, 2 and 3
    @VersionCode({1, 2, 3})
    public void foo() {}

    // bar() is called when the app is updated to version code = 4
    @VersionCode(4)
    public void bar() {}
}
```

Version Management

@KeithYokoma - Drivemode, Inc.

potatotips #11