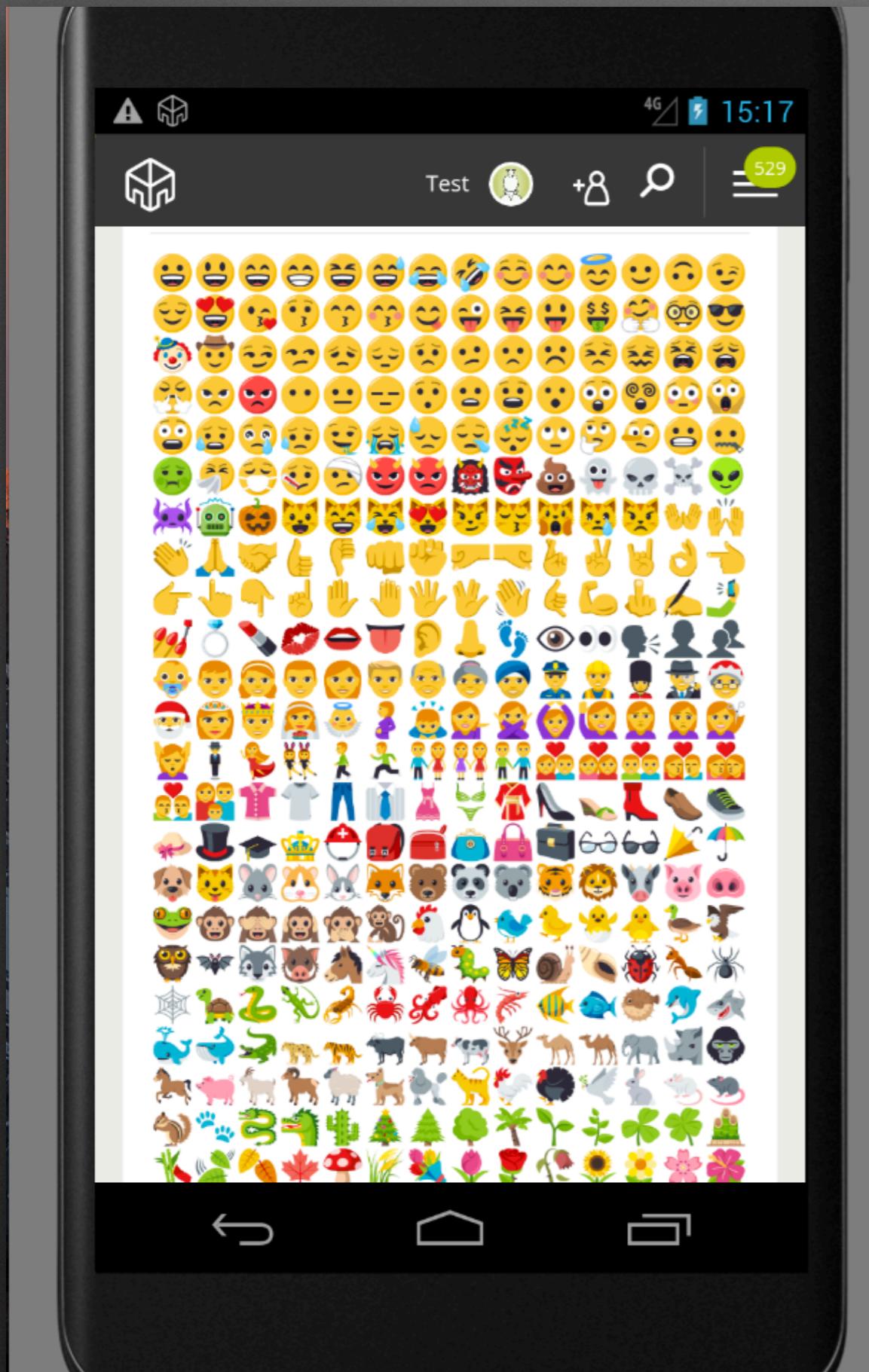
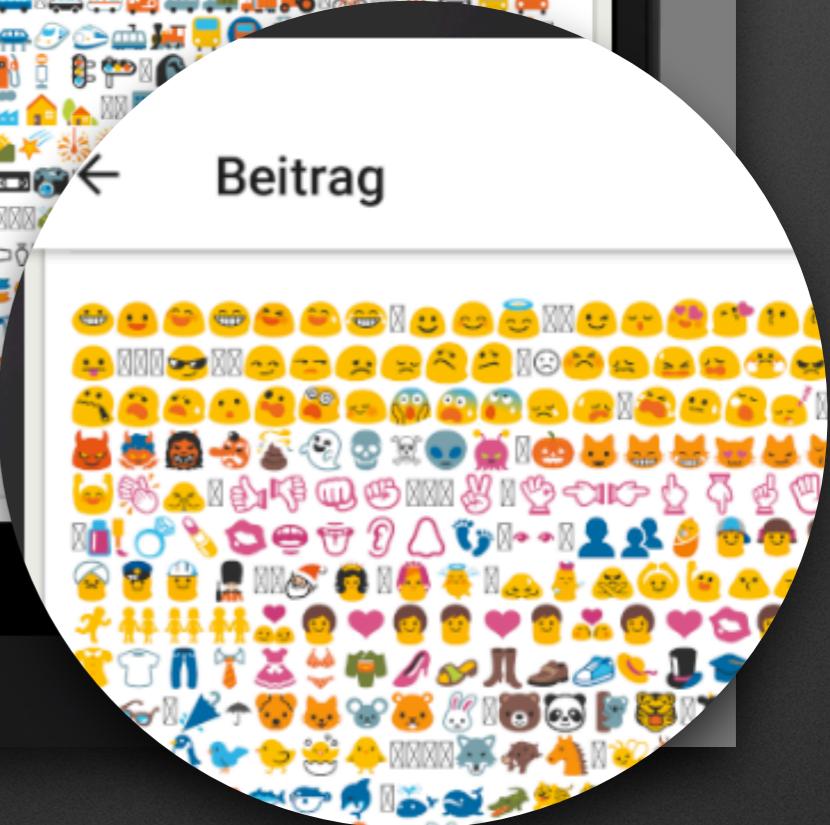
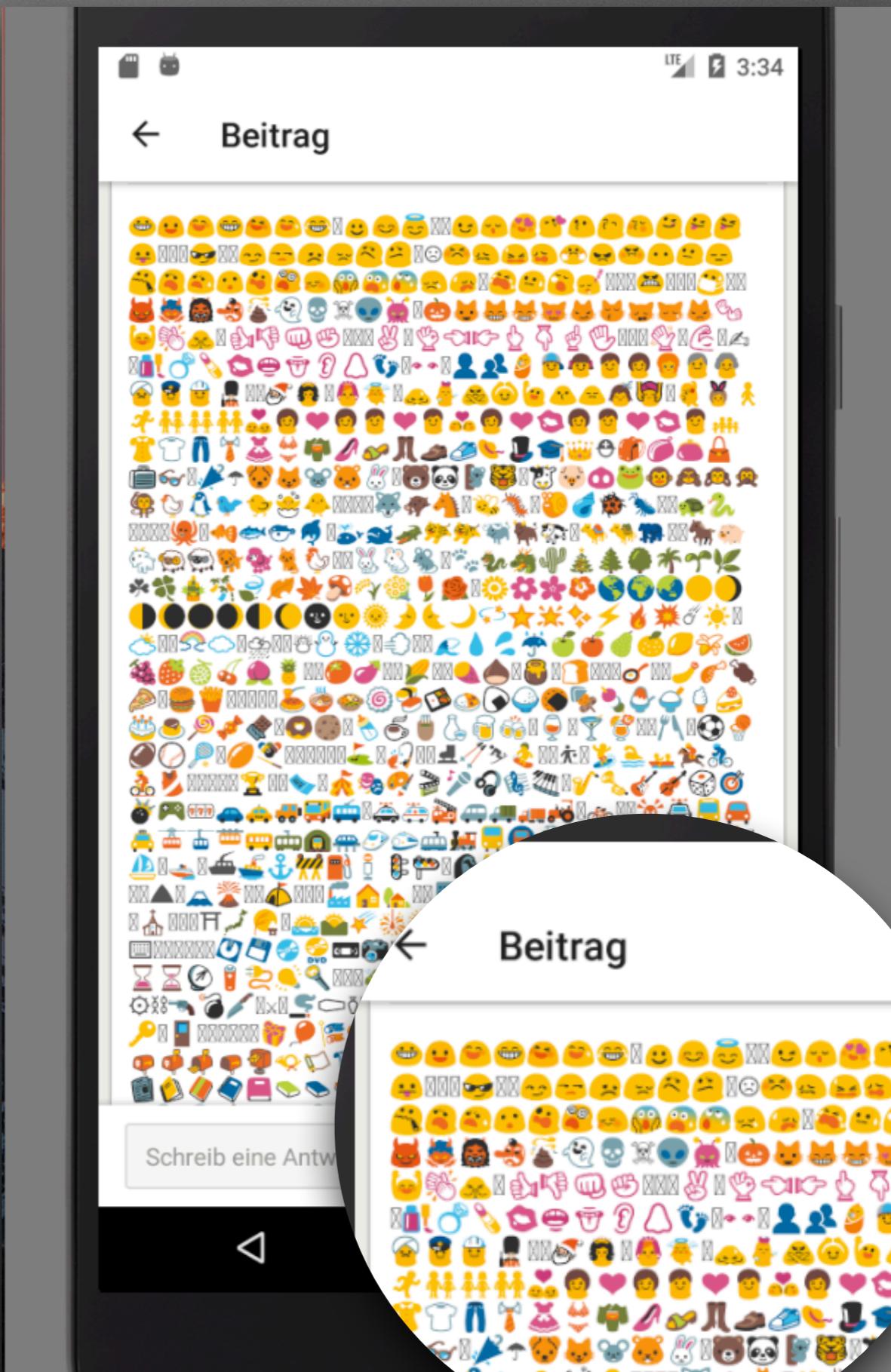
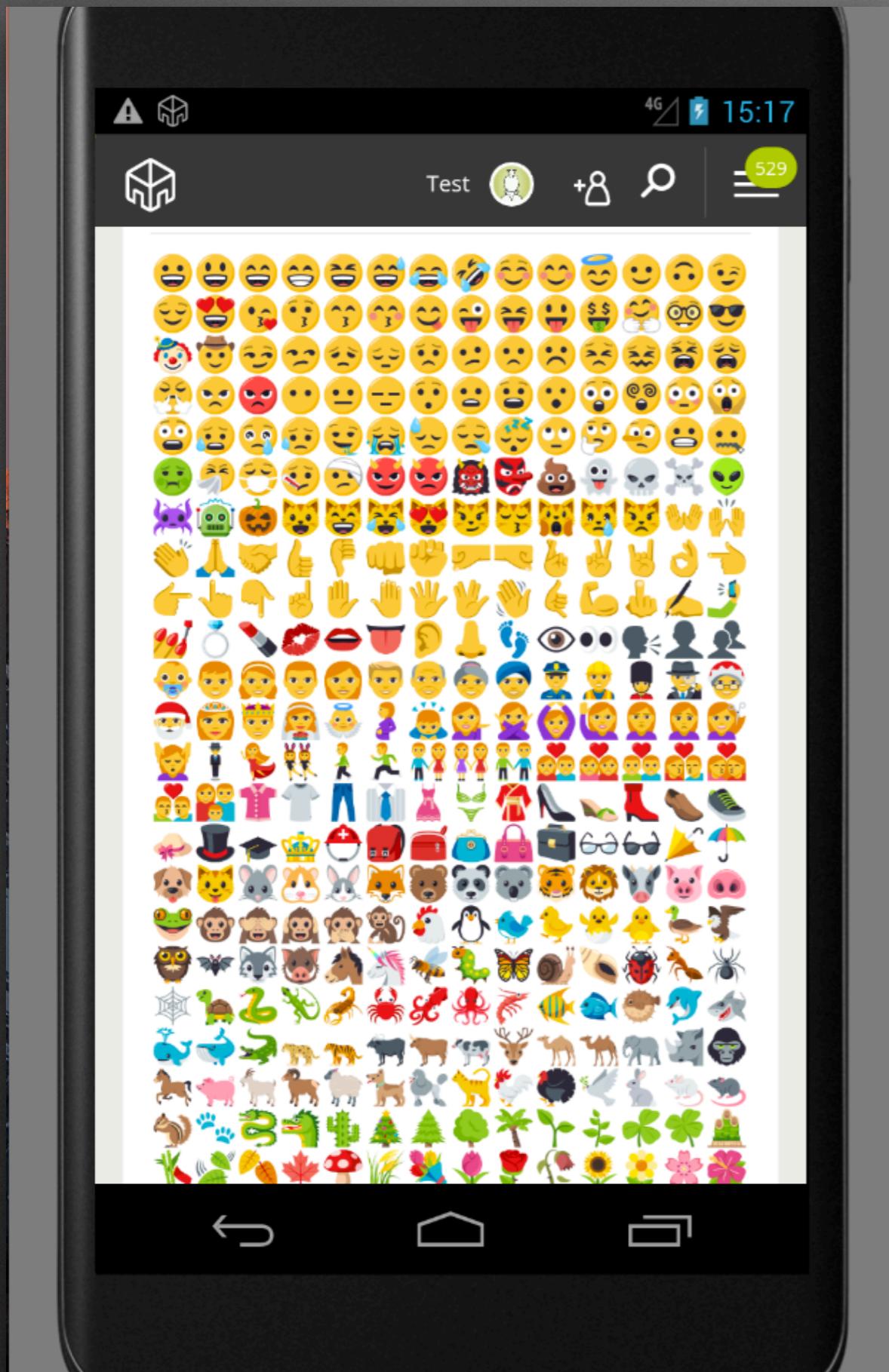


No More □

Taming Emoji on Android





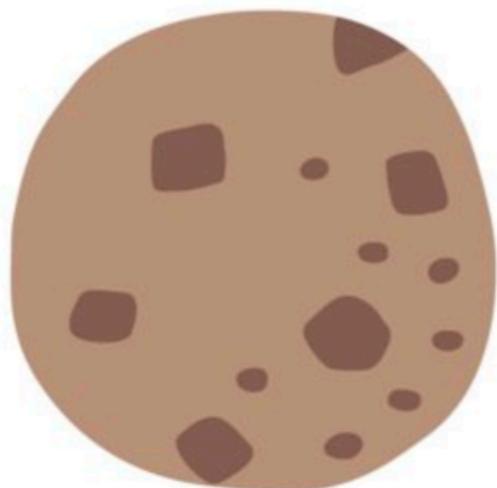
Apple



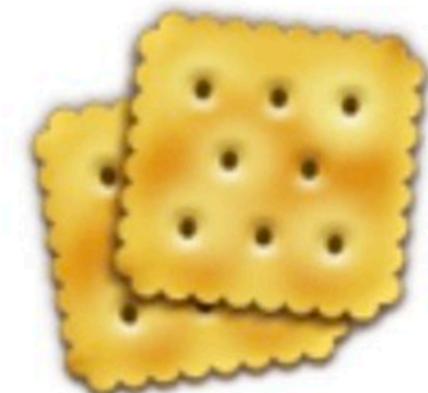
LG



Google



Samsung



Microsoft

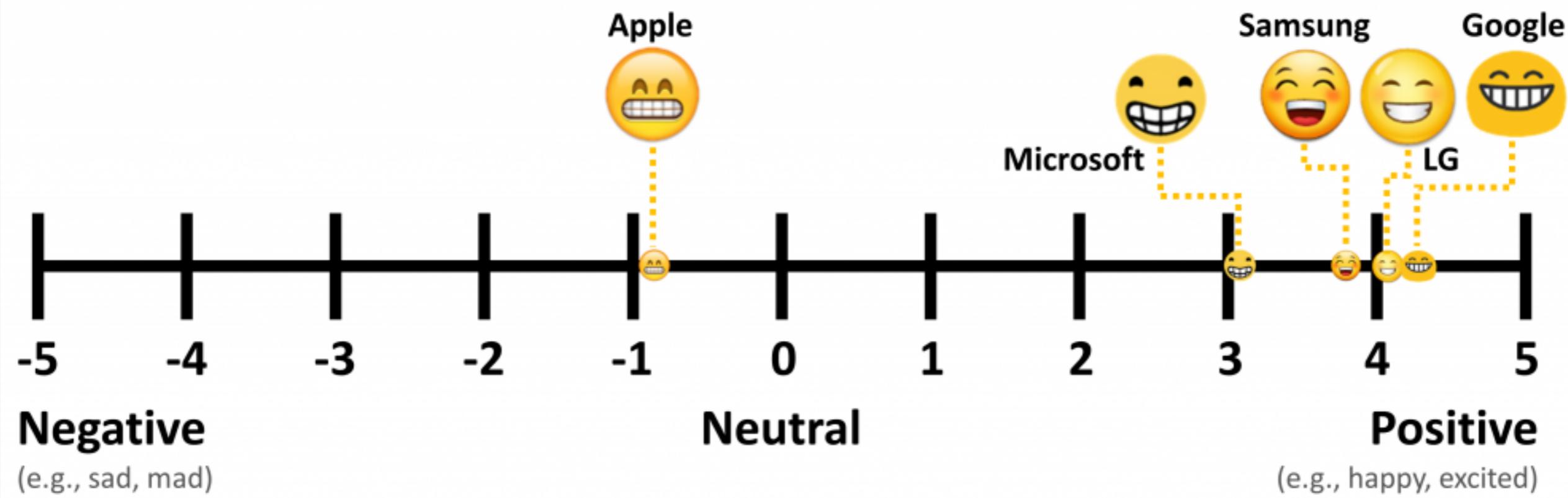


HTC



Same Emoji + Different Smartphone Platform = Different Emotion

For example, if you send the Apple emoji to a Google Nexus, they'll see the Google emoji, and vice versa!



Source: People interpret the same emoji in completely different ways

Link: <https://goo.gl/cxSFFq>

How Emoji works

```
val emoji = "\uD83D\uDE02"
```



U+1F469

U+200D

```
val emoji2 = "\uD83D\uDC69\uD83C\uDFFE\u200D\uD83D\uDCBB"
```



U+1F3FE



U+1F4BB



How long is an Emoji?

Problem: Users must write at least two characters.

```
val emoji2 = "\uD83D\uDC69\uD83C\uDFFE\u200D\uD83D\uDCBB"
```

```
emoji2.length = 7
```

```
Character.codePointCount(emoji2, 0, emoji2.length) = 4
```

```
val it = BreakIterator.getCharacterInstance()
it.setText(emoji2)
var count = 0
while (it.next() != BreakIterator.DONE) {
    count++
}
```

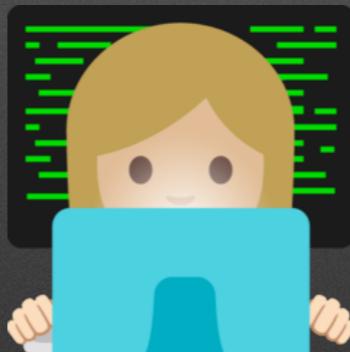
= 3 on API 23
= 1 on API 25

:shortname:

:joy:



:woman_technologist_medium_light_skin_tone:



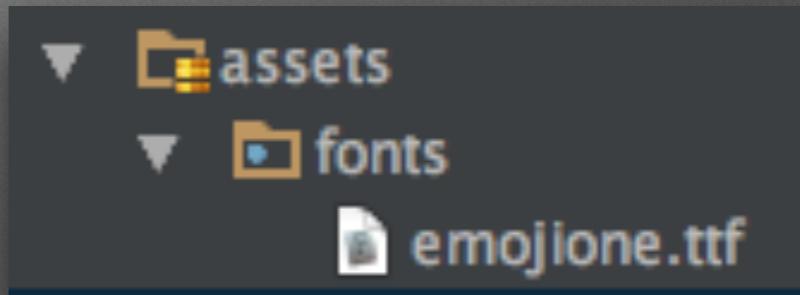
vdurmont/emoji-java
emojione

Problems so far:

- Missing Emoji (tofu)
- Different Emoji per platform
- Length of strings with Emoji
- Shortname <→ Unicode

EmojiOne custom font

Place in assets/fonts



```
val typeface = Typeface.createFromAsset(assets,  
                                         "fonts/emojione.ttf")  
textView4.typeface = typeface  
  
textView4.text = "Joy: $emoji\nDeveloper: $emoji2"
```

Joy: 😁
Developer: 🚧💻🖱️



Joy: 😂
Developer: 🚧💻🖱️

Download: <https://github.com/emojione/emojione/tree/master/extras/fonts>

Hold on...

```
Fatal Exception: java.lang.RuntimeException: native typeface cannot be made  
at android.graphics.Typeface.<init>(Typeface.java:321)  
at android.graphics.Typeface.createFromAsset(Typeface.java:295)
```

Problem: Fonts with color bitmap, FreeType >2.5

API < 23 Not supported!

Link: <https://github.com/emojione/emojione/issues/252>

Hold on...

```
textView4.text = "Numbers: 1 2 3 4"
```

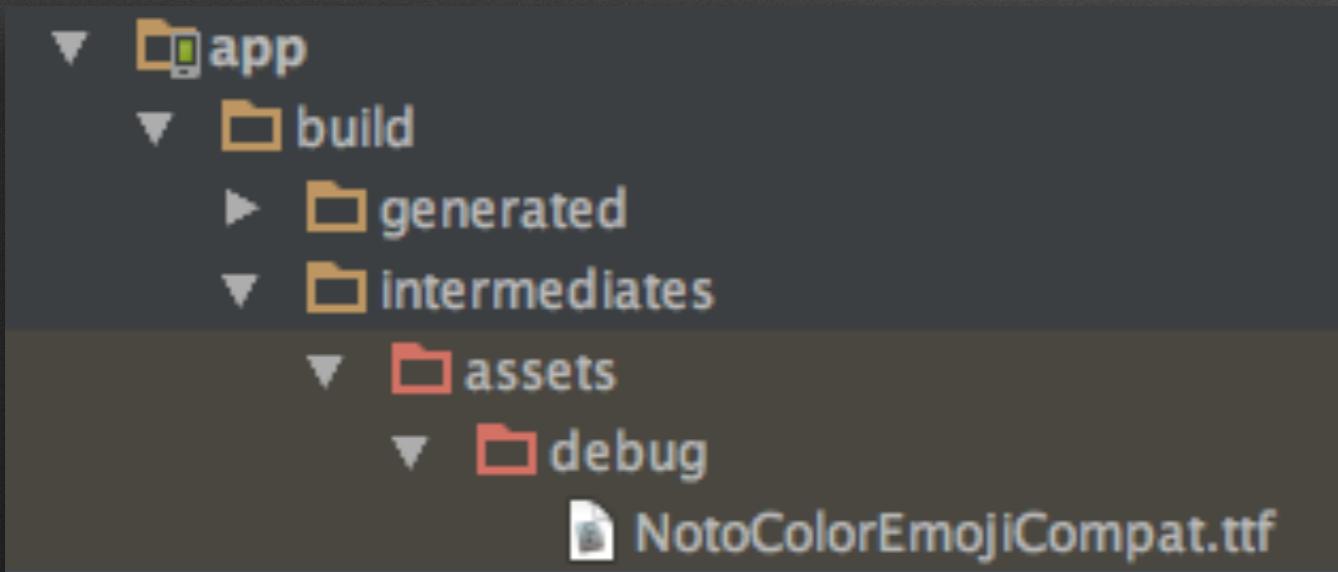
Numbers:  1 2 3 4

Solution: Parse Emoji & create TypefaceSpans

EmojiCompat in Action

Disclaimer: 26.0.0-Beta2 is broken! Slides use Beta1

```
compile "com.android.support:support-emoji:26.0.0-beta1"  
compile "com.android.support:support-emoji-bundled:26.0.0-beta1"
```



Adds: 7.4 MB

Init EmojiCompat

```
class MyApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        val config = BundledEmojiCompatConfig(this)  
        EmojiCompat.init(config)  
    }  
}
```

Takes 150 ms to init
(but happens asynchronously)

Using EmojiCompat

```
val string = "Joy: $emoji\nDeveloper: $emoji2"
val processed = EmojiCompat.get().process(string)
textView3.text = processed
```

```
▼  processed = {SpannableString@4268} "Joy:  \nDeveloper:  "
  ▶  f mSpanCount = 1
  ▶  f mSpanData = {int[27]@4275}
  ▼  f mSpans = {Object[9]@4276}
    ▶  ▷  0 = {TypefaceEmojiSpan@4278}
  ▶  f mText = "Joy:  \nDeveloper:  "
```

Joy: 😊
Developer: 🧑‍💻 💻



Joy: 😊
Developer: 🧑‍💻 💻

Replace all or not?

```
class MyApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        val config = BundledEmojiCompatConfig(this)  
        config.setReplaceAll(true)  
        EmojiCompat.init(config)  
    }  
}
```

Joy: 😂
Developer: 🧑‍💻



Joy: 😂
Developer: 🧑‍💻

Wait for it

```
EmojiCompat.get().registerInitCallback (   
    object : EmojiCompat.InitCallback() {  
        override fun onInitialized() {  
            val processed = EmojiCompat.get().process(string)  
            textView3.text = processed  
        }  
        override fun onFailed(throwable: Throwable?) {  
        }  
    })
```

Also available as View

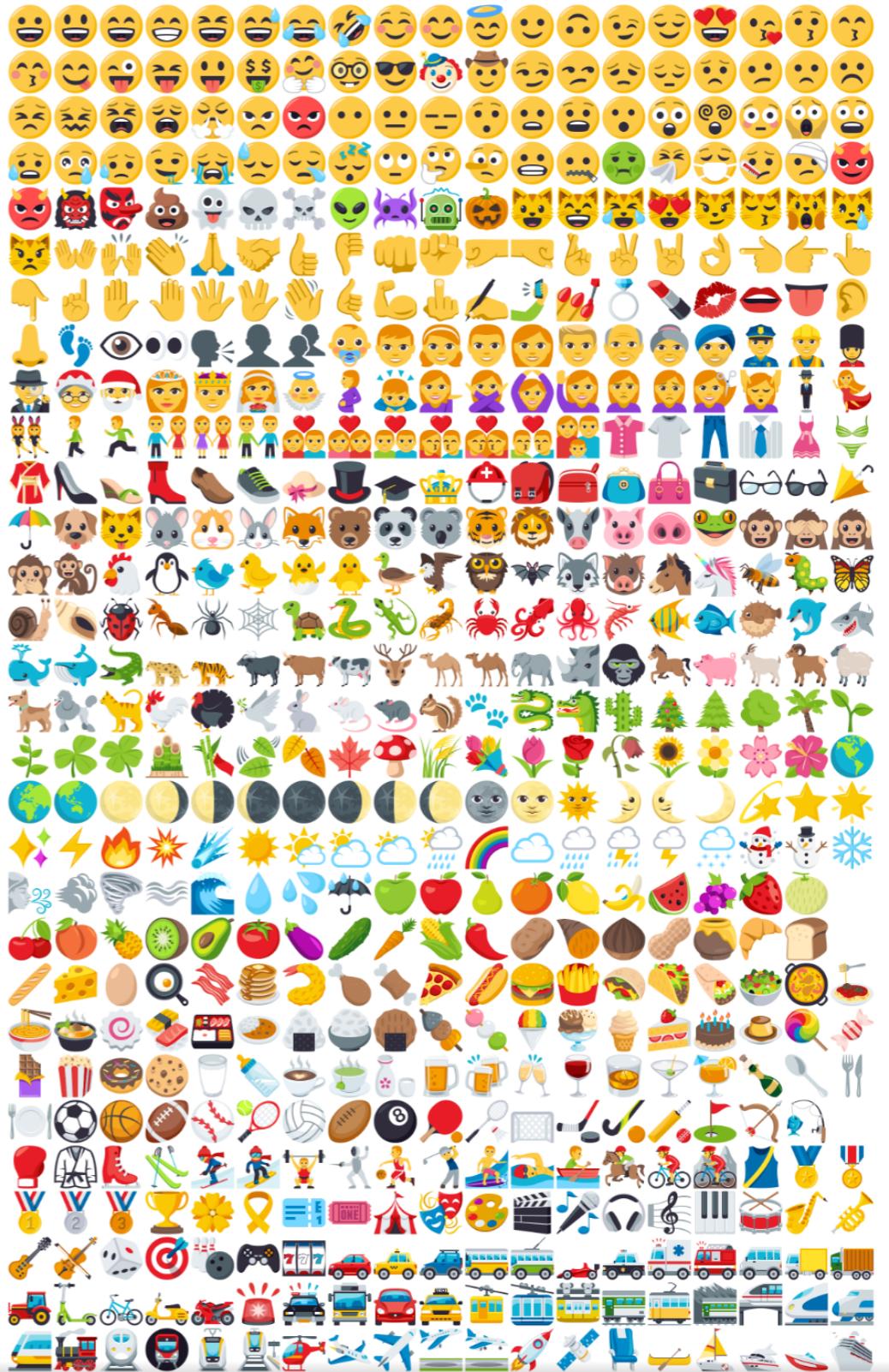
```
<android.support.text.emoji.widget.EmojiTextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>  
  
<android.support.text.emoji.widget.EmojiEditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>  
  
<android.support.text.emoji.widget.EmojiButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```



Beitrag



Beitrag



Thank you!



Code samples + slides:
<https://github.com/miquelbeltran/EmojiTalk>