Da-Jin Chu

12/17/15

B Section

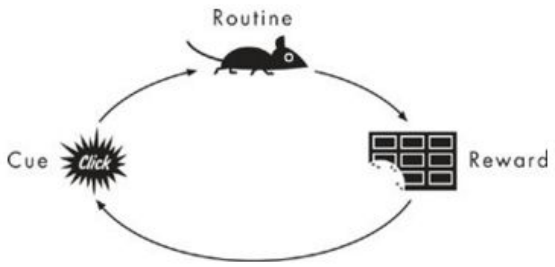
Introduction

Every new years, 45% of Americans make resolutions, but only 8% are successful after two years (Norcross, n.d.). The market for self-help books and blogs claiming to hold the keys to developing habits quickly and easily has grown, but very little of what is commonly claimed is backed in science. Frequently habits are oversimplified, with blogs claiming that 21 unbroken days of repetition would engrain any habit. In reality, however, habits are much more nuanced (Dean, n.d.).

Habits

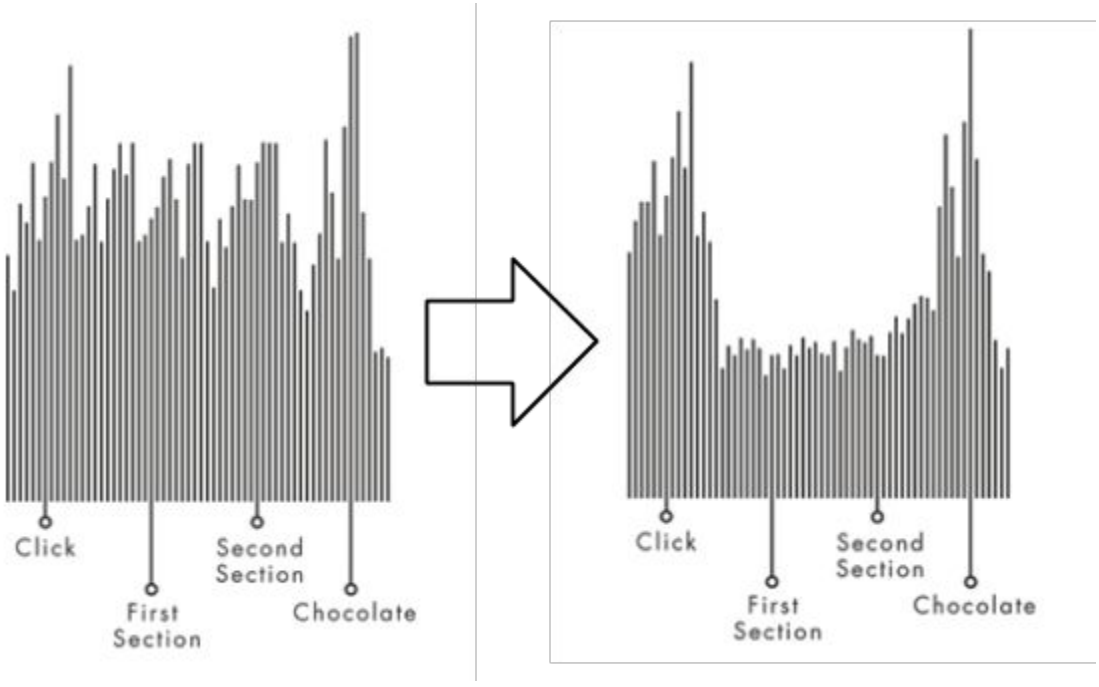
Habits are chunks of behaviors used by the brain to lessen its workload. Everyday, we rely on hundreds of these chunks to help us function without being overloaded by information and decisions. Each chunk consists of a cue, routine, and reward, called the habit loop. The cue causes the brain to begin the habit execution, and picks which routine to use. The routine is the actual activity of the habit, and the reward lets the brain know if the chunk should be remembered and used later. This process allows the brain to operate more efficiently because as the habit forms, less and less thinking is required to execute the routine. In addition, as the routine begins to require less brain power, the cue and reward each receive a spike in brain activity. This is because the brain ensures that is properly reading the situation it is in before launching into the unthinking and automatic routine. Depending on the cue, the brain will pick different routines to enter, which could be vastly different, so the brain works very hard at the

beginning of the habit loop to start the appropriate routine. At the end, when the brain encounters the reward, the brain needs to wake itself up and check that everything during the routine happened properly (Duhigg, 2012).



THE HABIT LOOP

As the habit loop is repeated, habits become stronger and more automatic. Brain activity before a habit is formed is high at all times as it processes new information and constantly makes new decisions. Over time, the dip in activity during the routine becomes deeper (Duhigg, 2012).



Brain activity of a rat running a maze before and after a week of repeated trials

The reduced brain activity during the routine occurs not only because the brain simply remembers what to do, but also because habits are executed in a distinct part of the brain, the basal ganglia. The basal ganglia is a very primitive structure located in the innermost layers of the brain. It is generally responsible for basic behaviors such as breathing and swallowing, whereas the outermost layers of human brains are responsible for complex thoughts like understanding jokes or making inventions. By moving repeated actions to the basal ganglia, the rest of the brain can sleep or work on other tasks while the basal ganglia recalls and acts on its learned patterns (Duhigg, 2012).

Changing Habits

Habits are never fully erased. Once a habit is formed in the basal ganglia, it is there indefinitely. This is helpful because skills don't need to be relearned after a break, but can be detrimental because the brain doesn't know which habits are good or bad. All habits are ready to be activated by their cues and rewards. This makes changing and creating new habits very difficult. Routines never leave our brains, and cues will always try to activate the rewards. However, new habit routines can be created to be stronger than old ones, using a new habit loop to overpower the old one. After the new habit is formed, the positive habit is just as automatic as any other (Duhigg, 2012).

Android Applications

The official integrated development environment to build applications in Android is Android Studio. Based on IntelliJ IDEA, Android studio features: a build system based on Gradle, a drag-and-drop graphical editor for app layouts, and the ability to sign apps, among many other features (Android Studio Overview, n.d.).

Android is a Linux based operating system. Each application installed is an archive file, known as an APK, or *Android package*, and has the .apk file extension. Apps are contained in APK files which are used to install the app onto a device. Once installed, each app is a different Linux user and runs on its own process and virtual machine. Separating apps into their own processes and users allows the Android system to better manage security. The system can grant permissions and access to data differently for each app, and start the app's process whenever needed (Android Fundamentals, n.d.).

Android follows the *principle of least privilege*, meaning that the operating system gives the minimum amount of access to components that is required for an app to work, and no more. Parts of the system can only be accessed by apps once the user has explicitly given permission, such as SMS messages, contacts, SD card storage, Bluetooth, and more. However, two apps can also choose to be part of the same Linux user, thus sharing the same process, virtual machine, and access to the same files. To do this the apps must have the same certificate sign them (Android Fundamentals, n.d.).

App Components

App components are the building blocks of applications. When any of an app's components needs to be run, the app's independent process and virtual machine will be started. Each component is separate and has a specific job, and together they define an app's behavior. Components may depend on each other and do not all have to be entry points for the user (Android Fundamentals, n.d.).

There are four types of components. Each has a unique purpose and process of being created and destroyed, known as a lifecycle. The four types are activities, services, content providers, and broadcast receivers (Android Fundamentals, n.d.).

Activities contain the user interface and are single screens. Activities might serve to show a list of emails, for example, or compose an email, or to read emails. Each would be a separate activity, and together, they create the user experience. Other apps can start activities, and if the email app allows it, other apps can choose to start of the activities (Android Fundamentals, n.d.).

Services run in the background. They are responsible for long-running or remote processes. Services do not have user interfaces. They may often be used to download data or play music without interfering with the user interface in or outside of the app that the service belongs to. Services can be started and left to run on their own, or bound to another component to allow the two to interact (Android Fundamentals, n.d.).

Content providers manage app data. Content providers allow other apps to read and sometimes write data, which can be stored in any location the provider's app has permission to access. The user's contact information, for example, is accessed through a content provider given by the Android system. Apps with permission can read and write the user's contact information with this content provider. Though primarily used to share data amongst apps and the system, content providers can also be useful for managing data within a single app (Android Fundamentals, n.d.).

Broadcast receivers catch system-wide announcements. Broadcast announcements are often from the system. For example, common announcements include the screen turning off, the battery getting low, or a picture being taken. Non-system apps can also broadcast

announcements. Broadcast receivers have no user interface and are most commonly used as a "gateway" leading to other app components, but may launch status bar notifications (Android Fundamentals, n.d.).

Android Manifest

The Android system cannot start app components without first knowing that the component exists. An app can tell the system about its components by declaring them in a file named AndroidManifest.xml which must declare all components, and be located at the root of the project directory. In addition to components, the manifest file also declares required permissions, Android API level, device hardware or software, API libraries such as the Google Maps library, and more (Android Fundamentals, n.d.).

Android apps also need resources other than source code. This includes images, audio files, or anything else used in the user interface. Layouts, menus, colors, or even animations are supposed to be defined with XML files. Using resources allows the developer to modify the app without touching any code, and instead simply using different resources. This is also very useful for supporting different combinations of languages and screen sizes (Android Fundamentals, n.d.).

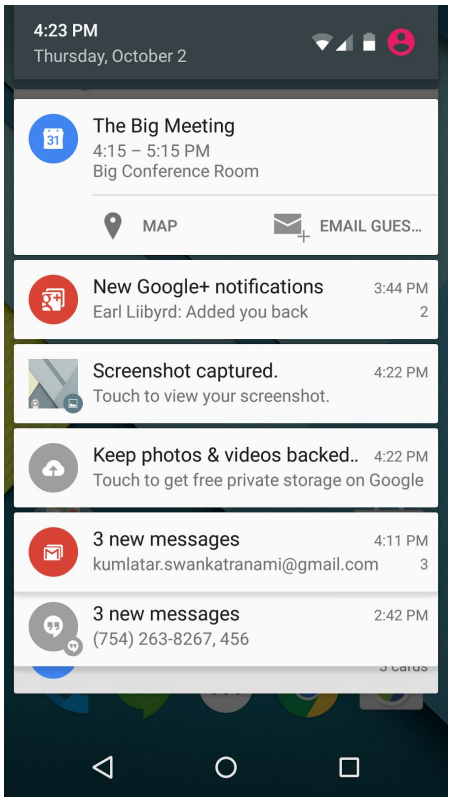
Every resource receives a distinct ID from the SDK build tools, which is used to refer to that resource from source code or other resources (Android Fundamentals, n.d.).

Android allows developers to use a short string appended to the name of directories, called a qualifier, which defines which types of devices should use the resources in that directory, allowing developers to support different screen sizes, pixel densities, languages, and other configurations, all without touching any code (Android Fundamentals, n.d.).

Notifications

Notifications are messages sent to the user. They are not part of an application's regular UI. When an app tells the system to send a notification, it pops up in the notification area as an icon first. The details of the notification can be seen once the user accesses the notification drawer. The drawer and area are controlled by the system, allowing the user to see them no matter where they navigate. Notifications are very important to the user interface and have a specific set of design guidelines. Google's material design specifications should be followed (Notifications, n.d.).

Notification area

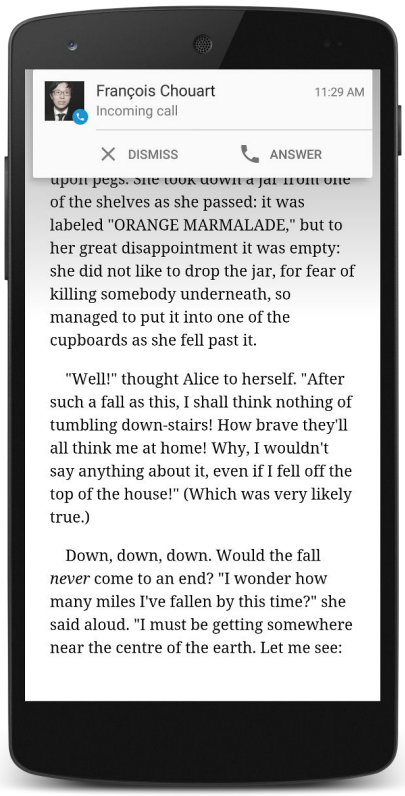


Android allows apps to add actions; these are shown on the bottom of the notifications. Actions let users perform the most used tasks for notifications from the notification itself without having to open the notification's application. Actions speed up the user's experience and allow the user to concentrate on what matters (Notifications, n.d.).

Developers need to be cautious with the number of actions added into a notification. Adding extra actions adds complications that can confuse the user. The number of actions should be restricted to the least number possible to just the urgent and important actions (Notifications, n.d.).

High-priority notifications are temporarily shown expanded to show their actions. The notification moves into the notification shade after some time (Notifications, n.d.).

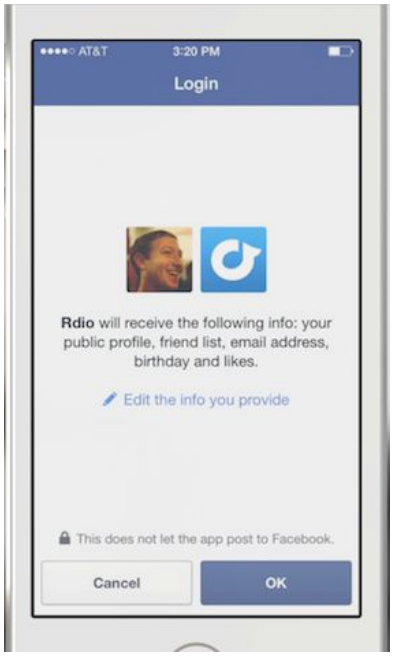
If its priority is High, Max, or full-screen, it will get a heads-up notification (Notifications, n.d.).



Sample heads-up notification

Facebook API

Facebook Login for Apps allows people to create accounts and login to an app quickly and easily from multiple platforms. It works on iOS, Android, WDeb, Windows Phone, as well as through desktop apps and other devices (Facebook Login, n.d.).



Facebook Login from iOS

Facebook Login allows users to rapidly make an account for an app without needing a password, reducing the chances of them forgetting passwords. It is a basic and easy process which brings better conversion rates to an app. By creating an account in one account, they can easily log into the app on any other platform. Email addresses that have been validated can be used to re-engage the owner of that email later (Facebook Login, n.d.).

Personal experiences engage users and retain users more. Facebook Login allows developers to gain access to information from Facebook that would be otherwise difficult to collect. This includes things they like, their birthday, hometown, location and work history.

Using the user's Facebook profile picture alone connects the app to the user more closely (Facebook Login, n.d.).

Apps that retain users allow people to make connections with their friends to experience the app together. Facebook Login lets the developer access which users are friends on Facebook so that they can be connected, thus building app value (Facebook Login, n.d.).

By logging into Facebook, people can share the actual identity and public profile, which includes the person's name, picture, gender and locale. Apps which use real identities are not plagued with as much spam and noise, allowing quality conversations to grow (Facebook Login, n.d.).

Facebook Login works on the most frequently used mobile and desktop platforms. People who make accounts through Facebook on a platform can easily login on a different account. People are known by the same User ID on any platform, so they can continue their experience wherever they left off (Facebook Login, n.d.).

The Graph API is used to get data in and out of Facebook. It is based on low-level HTTP that can perform tasks such as query data, post stories, manage ads, upload photos, and more. The design of the Graph API is based on the idea of the 'social graph' which represents the information on Facebook. It is composed of nodes, edges, and fields. Nodes can be anything, such as users, photos, pages, or comments. Edges are connections between nodes, like a Page's photos or a Photo's comments. Fields are information about things, like a user's birthday or the title of a Page (Graph API, n.d.).

Engineering Proposal

Engineering Problem:

Forming healthy habits is very difficult, especially when one lacks other people to keep them accountable.

Engineering Goals:

The goal is to develop an Android application that helps people build habits of their choosing by using notifications to remind users, and social media to hold users accountable.

Methods and Procedure:

Android Studio will be used to write the app. Every iteration will be recorded through GitHub. The first iterations will catch up with existing habit apps on the market. They will allow habits to be recorded daily. Afterwards, social media will be supported, then notifications, and then educational instructions for habits. Data will be collected to see if the added features that are not found in current habit apps impact habit building more. Beta testers will be found through Google Play Store's beta testing programs. This system keeps their identity anonymous by allowing testers to download the app themselves through the Google Play Store. Google Analytics data collected from the beta testers will be compared with habit building progression from users with or without the app's additional features. Google Analytics reports data anonymously through the app, and then analyze or export the data through the online console. The app should allow the user to record habit compliance daily. The app should educate users about how habits are formed, and inform the user on the best ways to build their chosen habits. Additionally, it should post progress data to social media. The app will be tested by giving a beta testing group multiple versions of the app. Some will have social media integration, some will

have habit building educational information, and some will have neither. Google Analytics will collect habit data.

Lit Cited:

Android Studio Overview. (n.d.). Retrieved November 1, 2015, from http://developer.android.com/tools/studio/index.html

Application Fundamentals. (n.d.). Retrieved November 1, 2015, from http://developer.android.com/guide/components/fundamentals.html

Dean, J. (n.d.). Making habits, breaking habits: Why we do things, why we don't, and how to make any change stick.

Duhigg, C. (2012). The power of habit: Why we do what we do in life and business. New York: Random House.

Facebook Login - Features Overview - Documentation - Facebook for Developers. (n.d.). Retrieved November 23, 2015, from https://developers.facebook.com/docs/facebook-login/overview

Graph API: Overview - Documentation - Facebook for Developers. (n.d.). Retrieved November 23, 2015, from https://developers.facebook.com/docs/graph-api/overview/

Norcross, J., Mrykalo, M., & Blagys, M. (n.d.). Auld lang Syne: Success predictors, change processes, and self-reported outcomes of New Year's resolvers and nonresolvers. Journal of Clinical Psychology J. Clin. Psychol., 58, 397-405. doi:10.1002

Notifications. (n.d.). Retrieved November 1, 2015, from http://developer.android.com/guide/topics/ui/notifiers/notifications.html

Notifications. (n.d.). Retrieved November 1, 2015, from http://developer.android.com/design/patterns/notifications.html