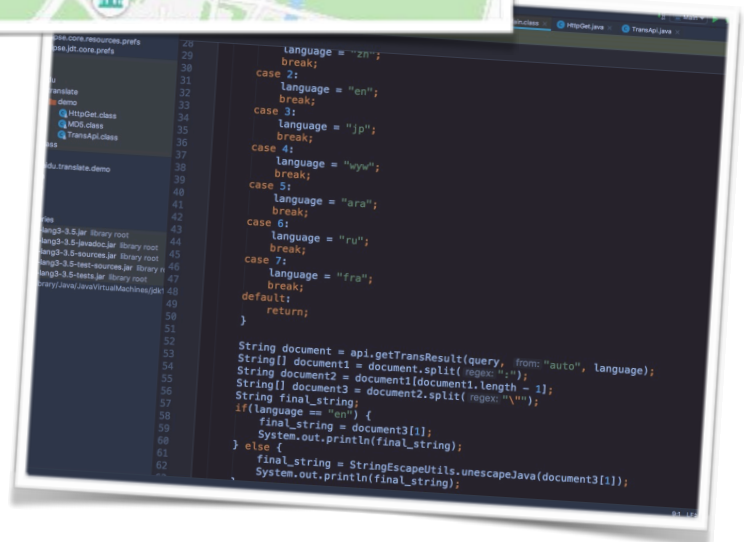
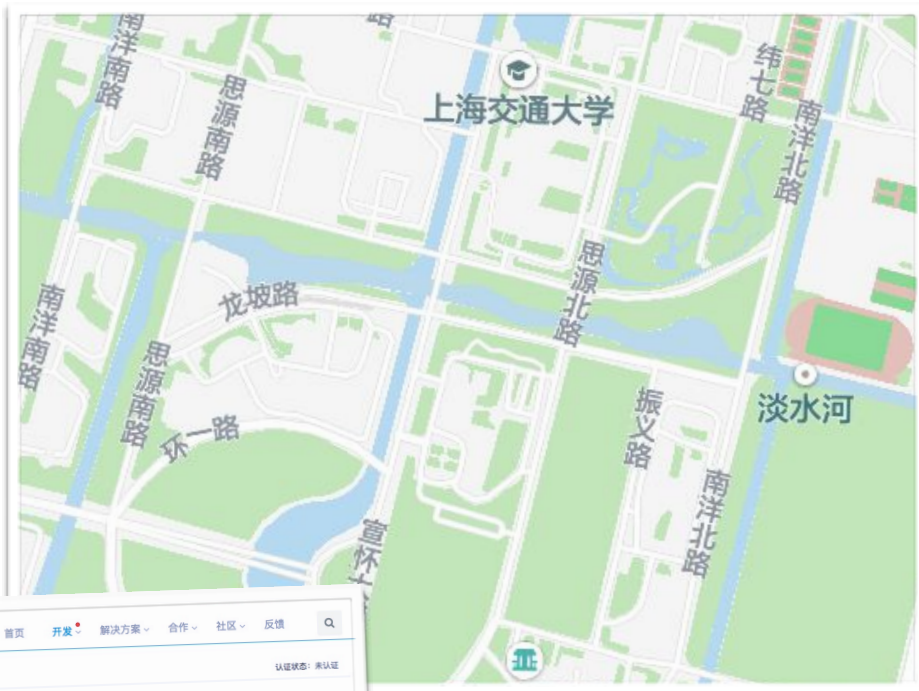


WORD MAP

REPORT



Word Map

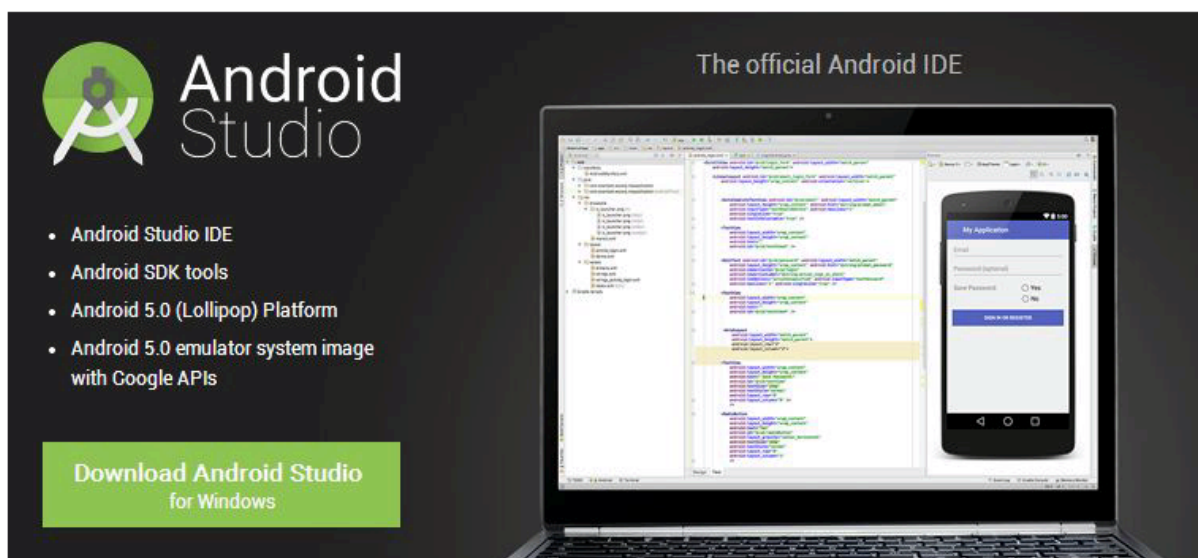
Introduction:

While English becomes increasingly essential these days, how to remember words became an important case in the process of learning English. Every person have a way of doing it and our project intends to introduce a new way. Studies have shown that people remember something better when it is provided with more hints. Thus, our group tried to associate words with places -the ones you have met with the words- to help learners remember words better. In our application, when clicking open a place on the map, words will appear, telling you what words you have noted down at this place.



Implementation:

The tool we used for the application is **Android Studio(AS)**. In the process of creating applications, UI is often closely connected with the code, while breaking the two parts up can cause later combination to be more difficult. However, in this



report, we will explain in the separately in order to make ourselves clearer.

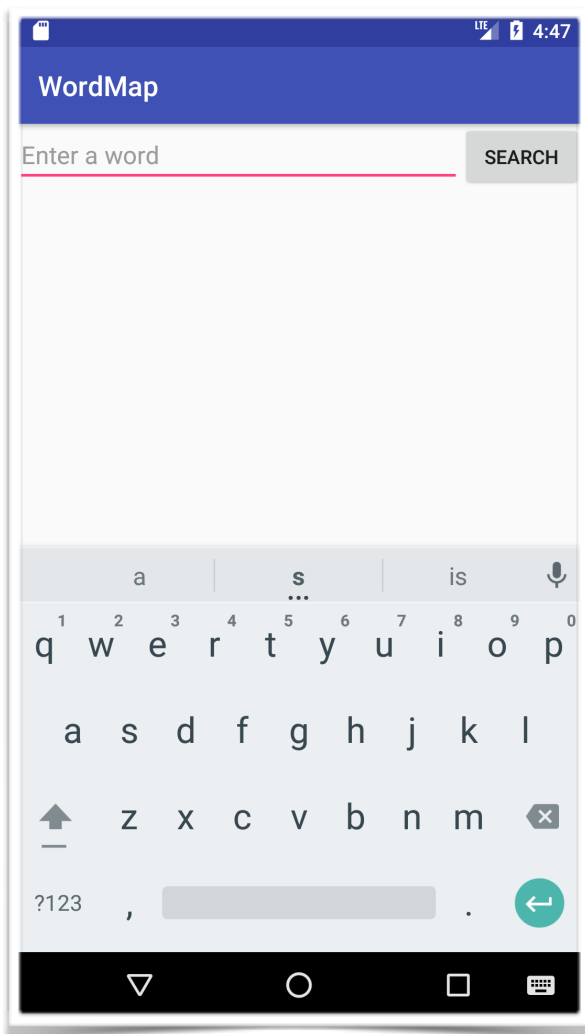
▶ UI-Mengqi Cao, Hao Xia

about UI

Android comes with some great tools to create and test applications, even if you don't have an Android device handy. That said, there's no replacement for actually touching your application. It's part of what makes Android such a compelling platform, the way it feels and responds (and the emulator just doesn't convey that).

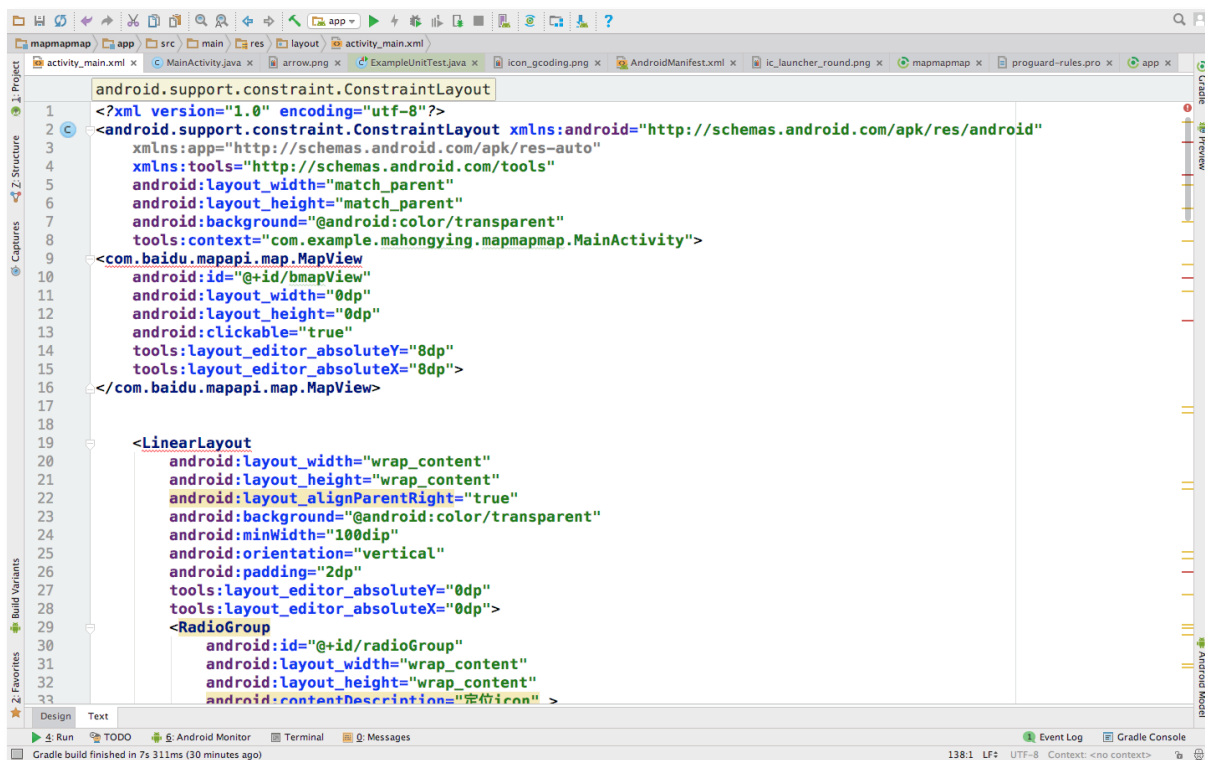
One of the most important tools in an Android developer's arsenal is the resource selection system. With it you can build highly dynamic applications that respond to changes in the devices, and thus, the user environment. Changing the screen layout based on the orientation of the device, or when the user slides out the phone's QWERTY keyboard, lets them know that you've taken their preferences into account when building your application. When building user interfaces in Android, it's strongly recommended to build at least the layout structure in an XML file. The XML layout files are not only considered as application resources, but Android also strongly favors building XML user interfaces over writing Java code. Sometimes, however, a layout XML file isn't enough, and you need to build parts of the user interface in Java. In this case it's a good idea to define at least a skeleton layout as XML(if possible), and then place the dynamically created View objects into the layout by using marker IDs and containers (much like dynamically adding to an HTML document in JavaScript).When building a user interface,

think carefully about the look and feel of the outcome. In our example, we use Button objects for the answers to questions. We could have used RadioButton objects instead, but then the user would have needed to select an option, and then touch a Next Question button, requiring two touches. We could also have used a List(which interacts nicely with the fact that it needs to be dynamically populated), however, a List doesn't indicate an "action" to



the user quite the way a Button does. When coding layouts, be careful with the measurement units that you use. It's strongly recommend that you stick to using sp for most

purposes—if you can't use one of the special `fill_parent` or `wrap_content` values. Other values are highly dependent on the size of screen, and won't respond to the user preferences. You can make use of the resource selection process to build different screen designs for small, medium, or large screens. You could also define your own



```
android.support.constraint.ConstraintLayout
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:background="@android:color/transparent"
8   tools:context="com.example.mahongying.mapmapmap.MainActivity">
9   <com.baidu.mapapi.map.MapView
10    android:id="@+id/bmapView"
11    android:layout_width="0dp"
12    android:layout_height="0dp"
13    android:clickable="true"
14    tools:layout_editor_absoluteY="8dp"
15    tools:layout_editor_absoluteX="8dp">
16   </com.baidu.mapapi.map.MapView>
17
18
19   <LinearLayout
20    android:layout_width="wrap_content"
21    android:layout_height="wrap_content"
22    android:layout_alignParentRight="true"
23    android:background="@android:color/transparent"
24    android:minWidth="100dp"
25    android:orientation="vertical"
26    android:padding="2dp"
27    tools:layout_editor_absoluteY="0dp"
28    tools:layout_editor_absoluteX="0dp">
29     <RadioGroup
30      android:id="@+id/radioGroup"
31      android:layout_width="wrap_content"
32      android:layout_height="wrap_content"
33      android:contentDescription="@string/icon" >
```

measurement unit and base it on the screen size. Always think about how your user will interact with your application, and how much (or little) time they are likely to have with it. Keeping each screen simple and responsive keeps your users happy.

About Activity

In the world of Android, an Activity is the point at which you make contact with your users. It's a screen where you

capture and present information to the user. You can construct your Activity screens by using either: XML layout files or hard-coded Java.



About presenting data

Data display is one of the most common requirements of a mobile application, and Android has many different options available. The ListView is probably one of the most commonly used widgets in

the standard Android suite, and styling it allows it to be used to display varying amounts of data, from one line menu items to multi-line to-do notes.

The GridView is effectively a tabular version of ListView, and is well suited for presenting the user with icon views. Icons have enormous advantages over text, since they can be recognized much more quickly by the user. Icons can also take up significantly less space, and in a GridView, you could easily fit four to six icons in a portrait screen without making the user interface cluttered or more difficult to work it. This also frees up precious screen space for other items to be displayed. Building custom Adapter classes not only allows you to take complete control over the styling of the ListView, but also determine where the data comes from, and how it's loaded. You could, for example, load the data directly from a web service by using an Adapter which generates dummy View objects until the web service

responds with actual data. Take a good look at the default Adapter implementations, they will generally serve your requirements, especially when coupled with a custom layout resource.

Activities and intents

Implementing an Activity at the correct granularity is an important part of your user interface design process. Although it's not a graphical part directly, it defines how the system will interact with your application, and thus how the user will interact with it. It's a good idea to keep implicit intents in mind when structuring how your Activity will be started. Creating a generic Activity allows for other applications to integrate seamlessly with your own, effectively turning your new application into a platform for other developers to work with. An implicitly started Activity can be replaced or extended by another application, or it can be re-used in other application. In both cases, the user becomes free to customize your application in much the same way that they can customize the wallpaper image or theme. Always try and provide a single Activity implementation for each action the user might want to take, don't make an Activity do too many things in the same screen. A very good example of granularity is the "Contacts" application—there's a contact list, contact viewer, contact editor, and the dialler application.

When working with tabbed interfaces (as we did in the previous chapter), it's possible to specify the tab content as an Intent, effectively embedding the Activity in your application. I would strongly urge you to consider doing

exactly this when building a tabbed user interface, since it allows each tab to be re-used by your application far more easily, while also allowing third-party developers to create extensions to your interface, one tab at a time.

▶ Algorithm–Haoxuan Wang, Hongying Ma

Haoxuan Wang:

The **algorithms** are not complicated, but involving many online resources. There are functions of connecting UI and code, set for the interaction of user and computer. In the word-translating part, we tried to use Baidu's API for translating words but it turned out that Baidu's words-translation function is not so powerful. Later on, we turned to our sources but the strings generated were not easy to deal with and dealing with them would be time-consuming as well as tiresome both for us and for the phone. At last, we decided to show the web-page of YouDao directly, which is both efficient and effective. After all, a new dictionary like YouDao or Lingoes is not what we aimed to do, and creating a new dictionary is meaningless, thus, we simply "copy" the web-page over and show the results.

Hongying Ma:

Our project wants to use maps to classify the words we have marked, which serves as a basis for classification.

My task :

1.Setting up a map and mark the word on the map and its algorithm.

2.The algorithm of translation.

Method:

Map: I Applied API on Baidu map official website, and called the GPS and Network to get the current position, in order to be more accurate, we let the map refresh for 5 seconds. for words, We set up a class to create a lexicon and set up two data members :first is the marker (belong to Baidu class), used to record the position ,second one is a “structure”, include two members: word and it’s meaning.



Our algorithm of adding word: use the function distance under the Baidu class to set a circle, we first add the word, then we create a point on the map, and add the words, Next time we add the words ,if distance between first point and this point less than 100 meters,

we let two words to be in the same position. if distance greater than 100 ,we create new point to display on the map.

Algorithm of translation: when we began doing the project, we wanted to get the meaning for the web page, and intend to download the html file and use the operation of strings to get the translation, but after we got a html, i can't find what i wanted. so i searched on web and knew that the web content has been encrypted for safety, and can't be got directly. So we change the method, we found we can use Baidu translation API to get the translation , and then just deal with

