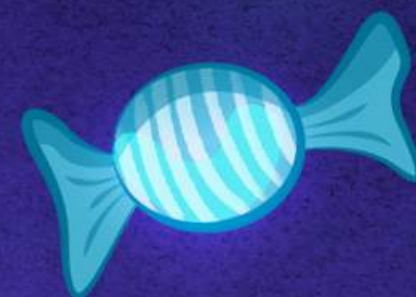
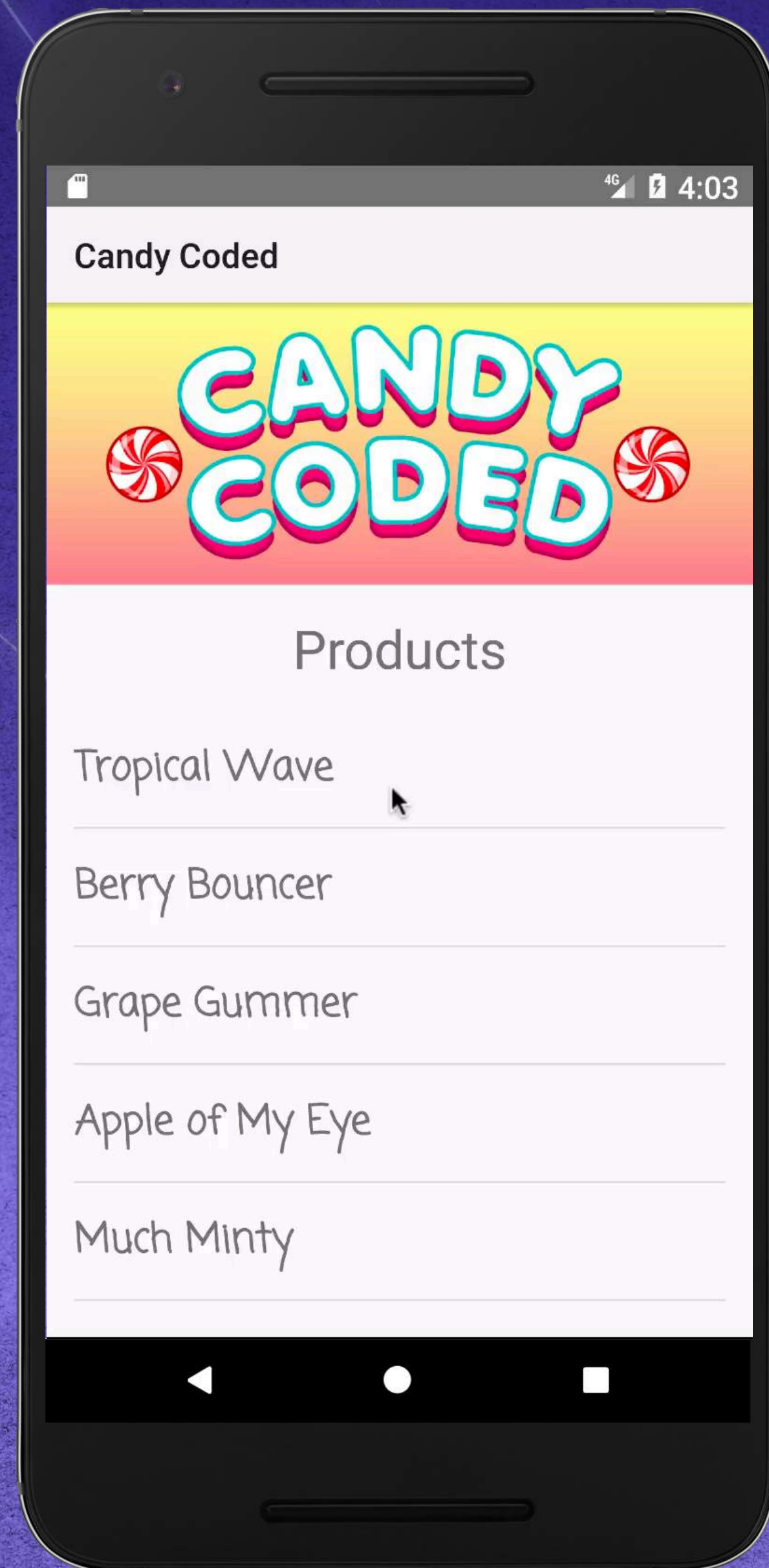


SUPER
SWEET
ANDROID
TIME



Where We Left Off in Try Android



This is the where the Candy Coded app left off in the Try Android course.



We created a candy store app and learned how to:

- Create an Activity **Layout**
- Add Images and an **ImageView**
- Add data to a **ListView with an Adapter**
- Create an **Event Listener**
- Display a **Toast**

SUPER
SWEET
ANDROID
TIME

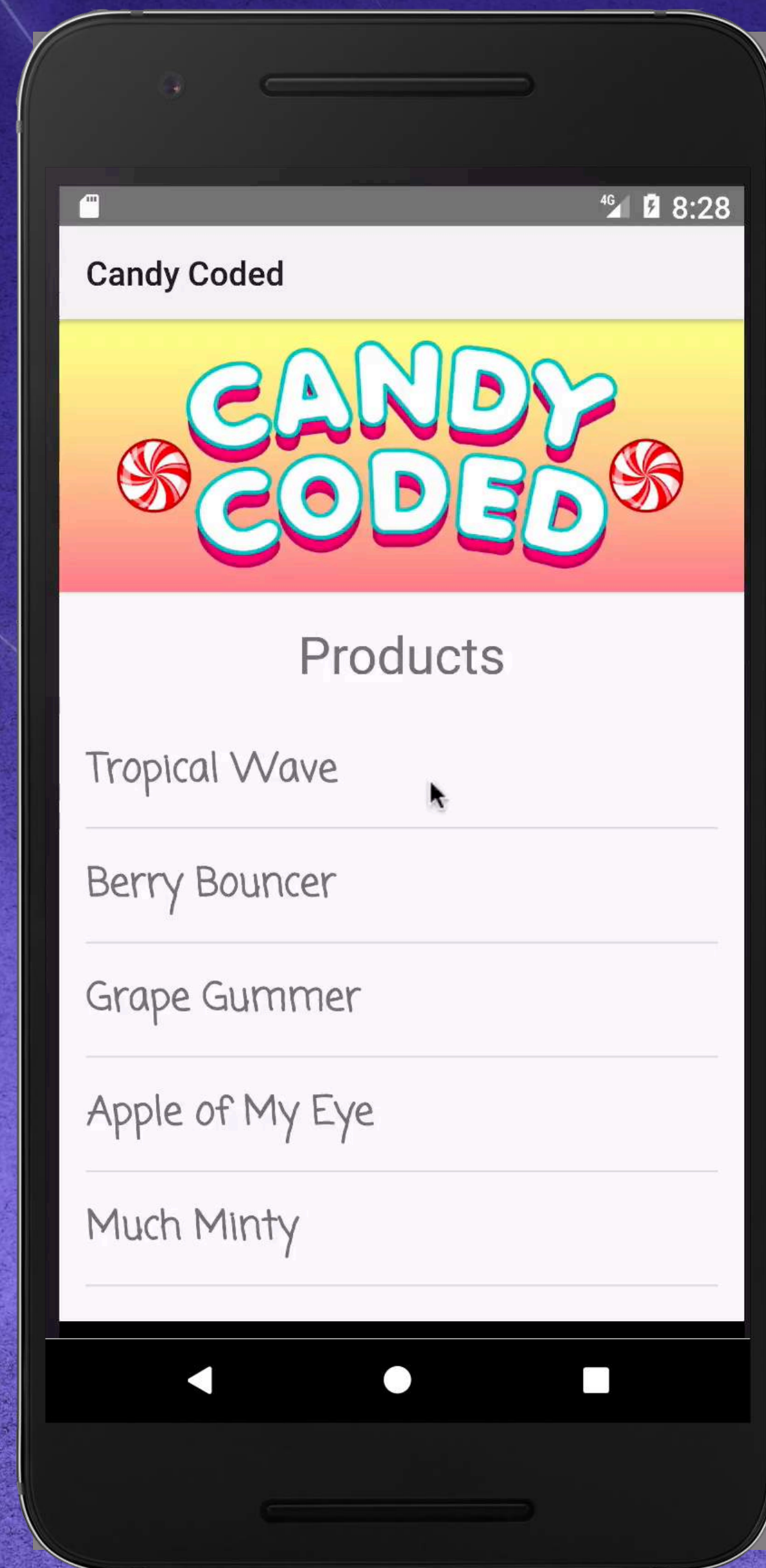


Screencast: Code Overview

SUPER
SWEET
ANDROID
TIME



The App We're Going to Build



In this course we're going to add more features to our Candy Coded app like:

- Launching another Activity **with an Intent**
- Performing a Network Request **with AsyncHttpClient**
- Parsing JSON into Objects **with GSON**
- **Designing a Constraint Layout**
- **Downloading and Cacheing Images with Picasso**
- **Storing Data in a Database**

You can download the finished Candy Coded app from the Try Android course here:
go.codeschool.com/TryAndroid-SampleApp



Level 1 Section 1

Activities and Intents

Launching an Activity with an Intent

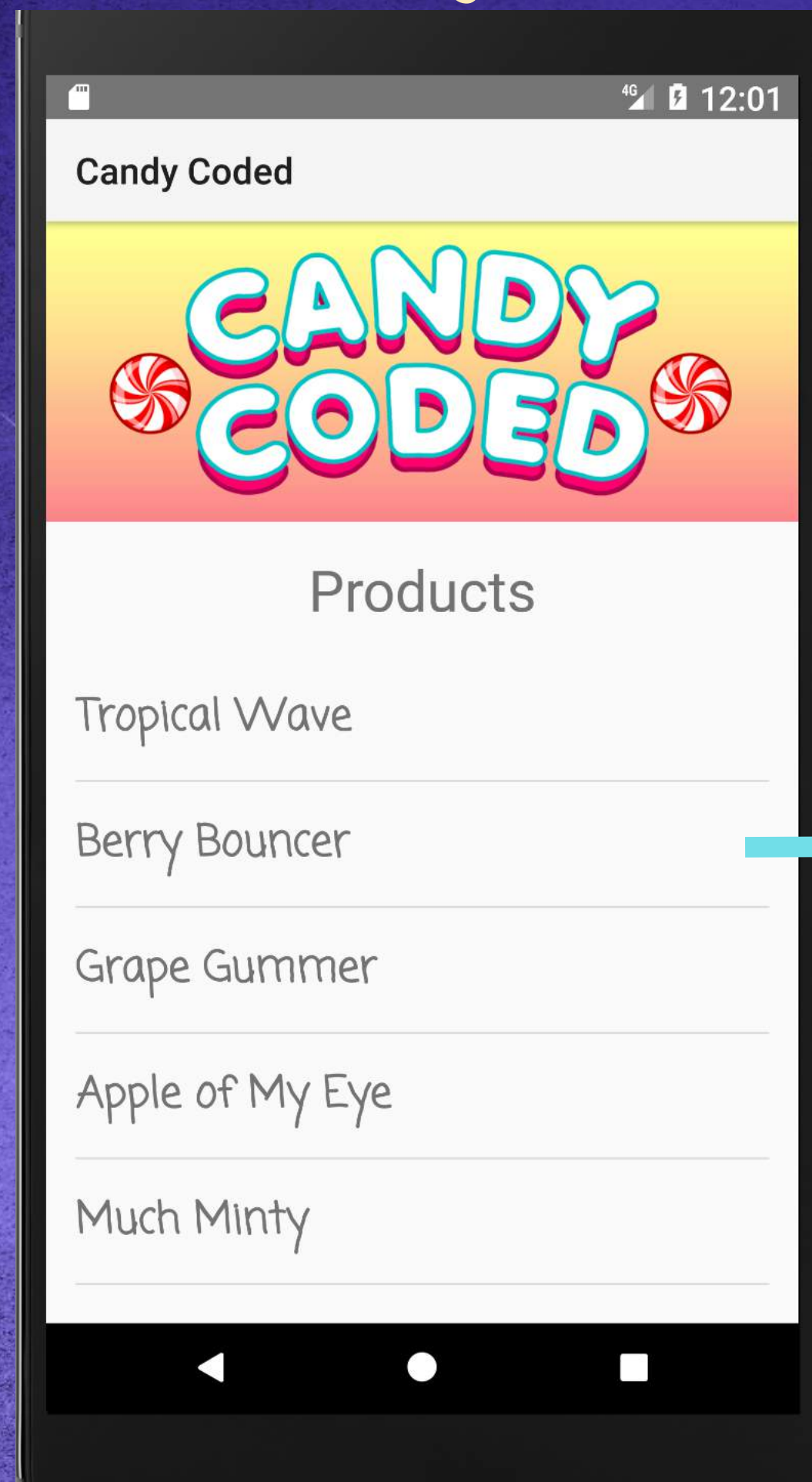
SUPER
SWEET
ANDROID
TIME



A New Activity will Show a Candy's Details

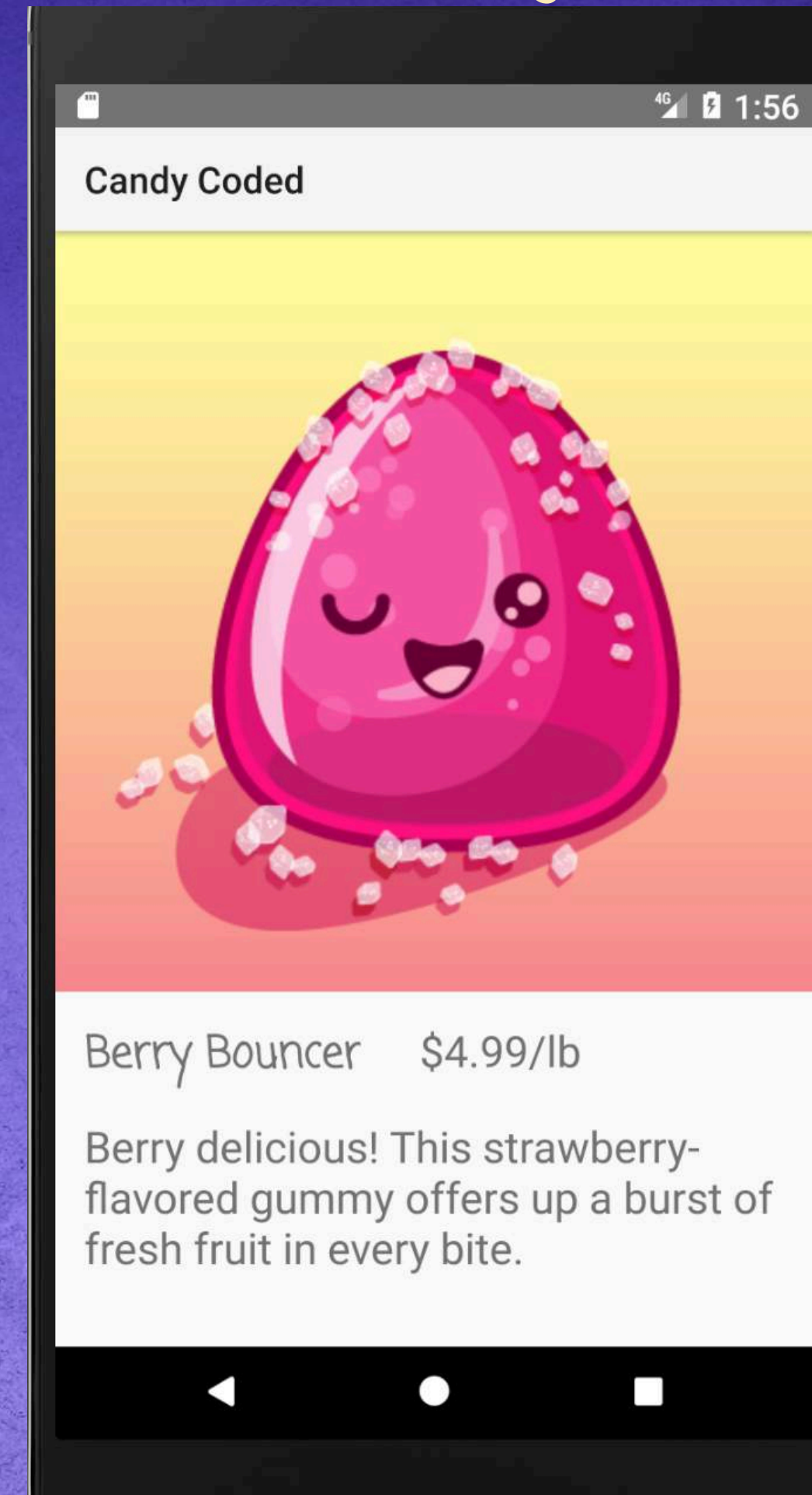
We want to create a new Activity called the DetailActivity to show the details of each candy.

Main Activity



*When a list item
is clicked the
Detail Activity
will start*

Detail Activity



*To do this we need
to create a new
Activity to show.*



Screencast: How to Create a New Activity

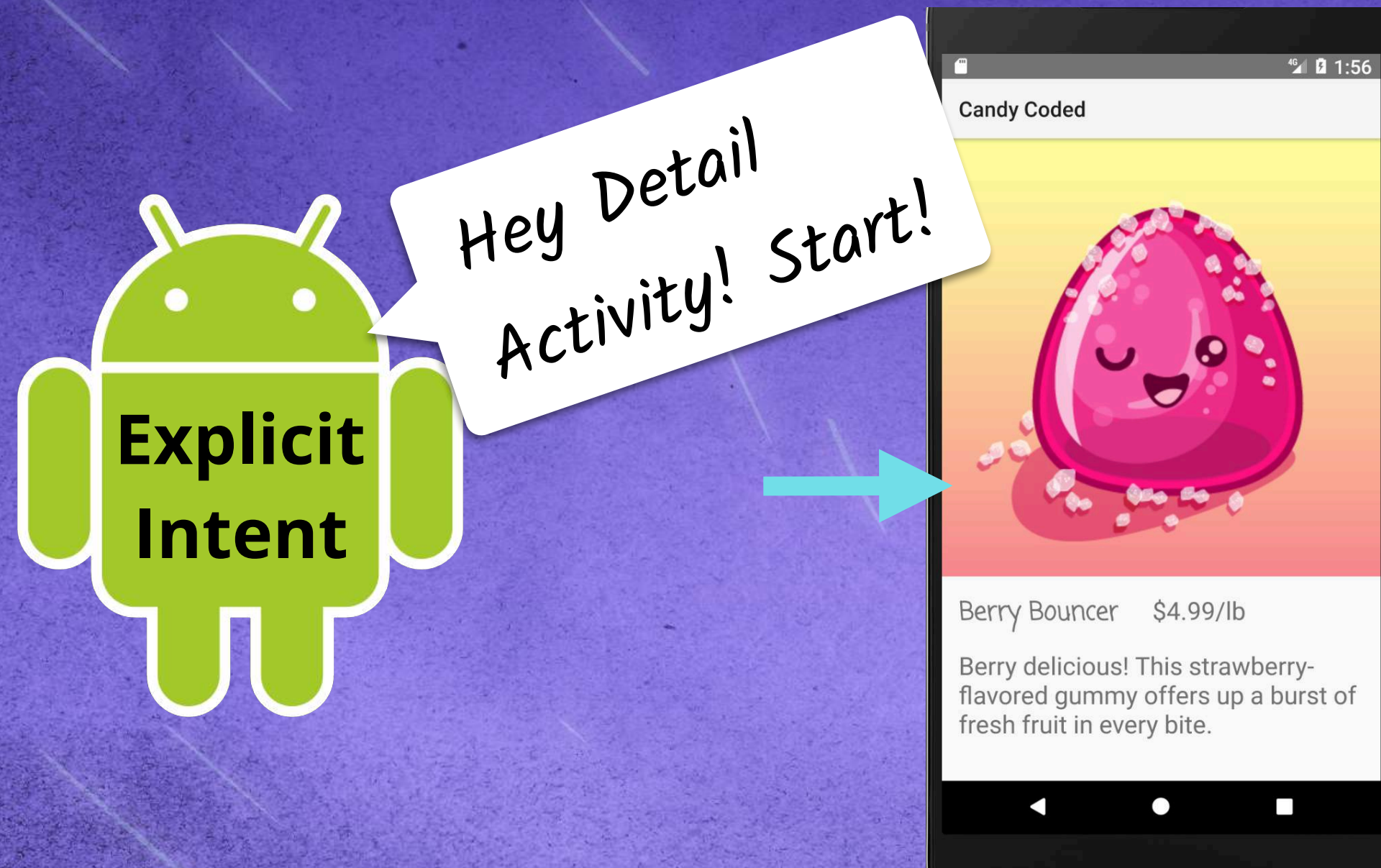
SUPER
SWEET
ANDROID
TIME



We Need an Intent to Launch Another Activity

An Intent requests an action from another app component.
There are two types of Intents - Explicit and Implicit.

Explicit intents launch specific app components like the Detail Activity in this case

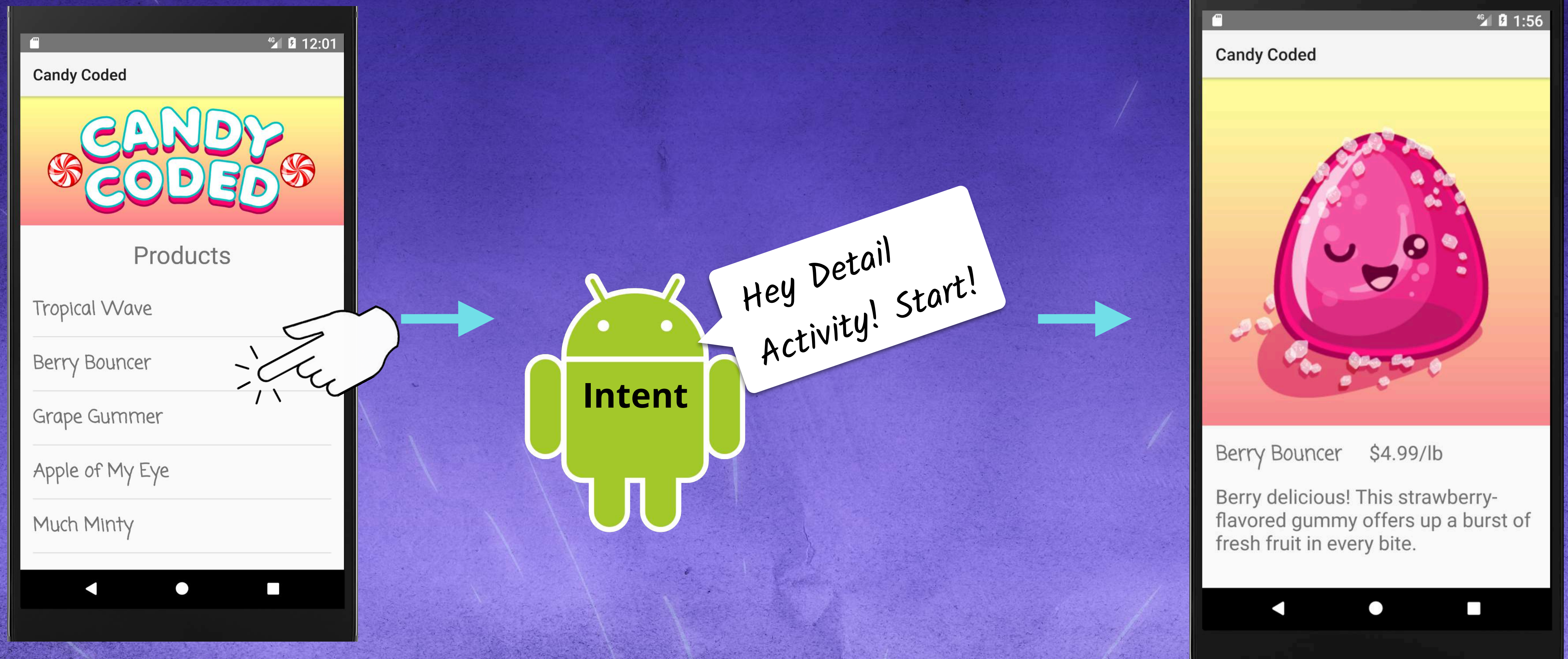


Implicit intents declare an action to perform, and the user can pick which app to use



We'll Use an Explicit Intent to Start the Detail Activity


When a User selects an item from our ListView our Item Click Listener will create an Intent to launch the Detail Activity.

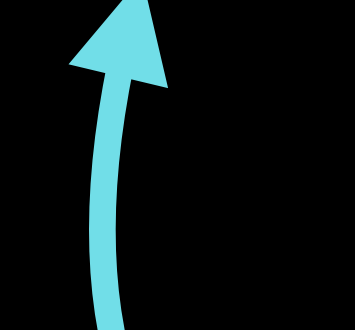


Creating the Intent in MainActivity.java

Since we want the Intent to start the Activity after a click we'll add that code to the Click Listener at the bottom of the onCreate() method in MainActivity.java.

MainActivity.java

```
...
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(
            AdapterView<?> adapterView, View view, int i, long l) {
            
        }
    });
...
```



Previously we were showing a Toast here, but we'll delete that code and only start our Detail Activity

Creating the Intent to Start the DetailActivity

To start an Activity we need an Intent created with the DetailActivity class we want to launch.

*The context of where we
are creating this Intent*

*The component used
in the Intent*

```
Intent detailIntent = new Intent(MainActivity.this,  
                                DetailActivity.class);  
startActivity(detailIntent);
```

*The MainActivity can start another Activity with the
startActivity() method, which takes an Intent as a parameter*

SUPER
SWEET
ANDROID
TIME



Creating the Intent in MainActivity.java

The Intent to start the Detail Activity **inside the** Click Listener **callback method** - onItemClick().

MainActivity.java

```
...
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(
            AdapterView<?> adapterView, View view, int i, long l) {
            Intent detailIntent = new Intent(MainActivity.this,
                                           DetailActivity.class);
            startActivity(detailIntent);
        }
    });
...
```


Screencast: Starting the DetailActivity

SUPER
SWEET
ANDROID
TIME



Level 1 Section 2

Activities and Intents

Passing Data with an Intent

SUPER
SWEET
ANDROID
TIME



How Can We Pass Data to an Activity?

We will need to pass data to our Detail Activity so that it knows what candy to show.



Passing Data to an Activity with an Intent

We can pass data in an Intent with the `putExtra()` method.

MainActivity.java

```
...  
Intent detailIntent = new Intent(MainActivity.this, DetailActivity.class);  
detailIntent.putExtra("candy_name", "Name of your candy here");  
startActivity(detailIntent);  
...
```

We need to assign a key to our data so we can look it up in the Activity

How do we know which candy item was selected from the list?

SUPER
SWEET
ANDROID
TIME



Passing the Selected Candy Name to the Detail Activity

Remember that our candy was stored in an `ArrayList`, so we can look up the candy to send to the Activity at position `i`.

MainActivity.java

```
...
final ArrayList<String> candyList = new ArrayList<String>();
candyList.add("Tropical Wave");
candyList.add("Berry Bouncer");
...
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(
        AdapterView<?> adapterView, View view, int i, long l) {
        Intent detailIntent = new Intent(MainActivity.this,
                                           DetailActivity.class);
        detailIntent.putExtra("candy_name", candyList.get(i));
        startActivity(detailIntent);
    }
});
...
```

Position in the List

Passing the candy in the Intent

Getting Data from an Intent in an Activity

In the Detail Activity, we can get the Intent and then get the data stored with it.

DetailActivity.java

```
...  
Intent intent = DetailActivity.this.getIntent();  
String candyName = intent.getStringExtra("candy_name");  
...
```

Get the Intent that created this Activity

Now we can look up the data inside the Intent with the `getStringExtra()` method and the key

Products
Tropical Wave
Berry Bouncer
Grape Gummer
Apple of My Eye



If Berry Bouncer was selected, then `candy_name` will be "Berry Bouncer" 🙌

SUPER
SWEET
ANDROID
TIME



Getting Data from an Intent in an Activity

In case the Intent doesn't have that key for some reason, we don't want our program to crash. We can check that it has that extra value first.

DetailActivity.java

```
...  
Intent intent = DetailActivity.this getIntent();  
String candyName = "";  
  
if (intent.hasExtra("candy_name")) {  
    candyName = intent.getStringExtra("candy_name");  
}  
...
```

Better to be safe than sorry!

SUPER
SWEET
ANDROID
TIME



Setting the TextView's text in DetailActivity

We have the `candy_name` that was passed in, so we can set the TextView's text to that candy.

DetailActivity.java

```
...  
Intent intent = DetailActivity.this getIntent();  
String candyName = "";  
  
if (intent.hasExtra("candy_name")) {  
    candyName = intent.getStringExtra("candy_name");  
}  
  
TextView textView = (TextView) this.findViewById(R.id.text_view_name);  
textView.setText(candyName);  
...
```

First we find the TextView by its id

Then we can set its text

Screencast: Passing Data to the DetailActivity

SUPER
SWEET
ANDROID
TIME



SUPER
SWEET
ANDROID
TIME

