Buse Çarık 06/07/18

INTERRA R&D Material Design

Material Design

Material Card

https://github.com/InterraMaterialDesign/MaterialCardView

**INDEX**

1. **INTRODUCTION……………………………………………………………………….3**
   1. **Purpose of this document…………………………………………………………...3**
   2. **Overview……………………………………………………………………………..3**
2. **WHAT IS CARDVİEW…………………………………………………………………4**
3. **ADVANTAGES AND DISADVANTAGES……………………………………………6**
4. **REQUIREMENTS FOR IMPLEMENTATION………………………………………6**
5. **AREAS OF USAGE……………………………………………………………………...6**
6. **MATERIAL CARDVİEW………………………………………………………………7**
   1. **Styles………………………………………………………………………………….7**
   2. **Attributes…………………………………………………………………………….7**
7. **EXAMPLES…………………………………………………………………………….10**
8. **REFERENCES………………………………………………………………………….15**

**INTRODUCTION**

**PURPOSE OF THIS DOCUMENT**

The aim of this project is analyzing material design card view’s style and attributes.

**OVERVIEW**

The project that is described in this document, includes material card’s style which is Widget.MaterialComponents.CardView.

**Card View**

CardView groups related information about a single topic in a container. Also, it can be used as summary or link for large details. CardView is extended from Frame Layout with shadow and rounded corners that are the main distinction between CardView and view groups.

CardView is needed a layout to hold and place all views in a card. The entire shape of the cards is called as container. Other views are optional; image views, text views, buttons and other views can be put into the cards. Buttons or image buttons can be a link for detailed information or another activity.

Usually, the CardView is used to represent a single item in a ListView or GridView cells. Before the CardView, each component that was used in the item, was implemented according to your design. For example, you can set an elevation for the card itself or you can make the entire content clickable by adding clickable feature to a card thanks to CardView. On the other hand, before the CardView, all these attributes were defined for every view in the cell.

Besides the inherited features from Frame Layout, CardView implements new attributes which are cardCornerRadius, cardElevation, cardMaxElevation, cardUseCompatPadding and cardPreventCornerPadding (These are explained detailly below, in Material CardView section). Rounded corners that are the one of the key differences, is obtained by cardCornerRadius attribute. The shadow effect is applied by the elevation feature. In a collection of cards, each card must be at the same elevation level. Another useful attribute is card background color which is set a color to the background of the card. In default, there are two different style options for background color which are CardView.Dark and CardView.Light. Light option is white(#ffffffff)

**\*\*\***CardView can be used earlier than the Lollipop with android.Support.v7.CardView package but it has different behavior. When the CardView is used before the Lollipop, it adds padding around the edges of the card and places the shadow between the border of the card and the additional padding area that is added. That padding is calculated

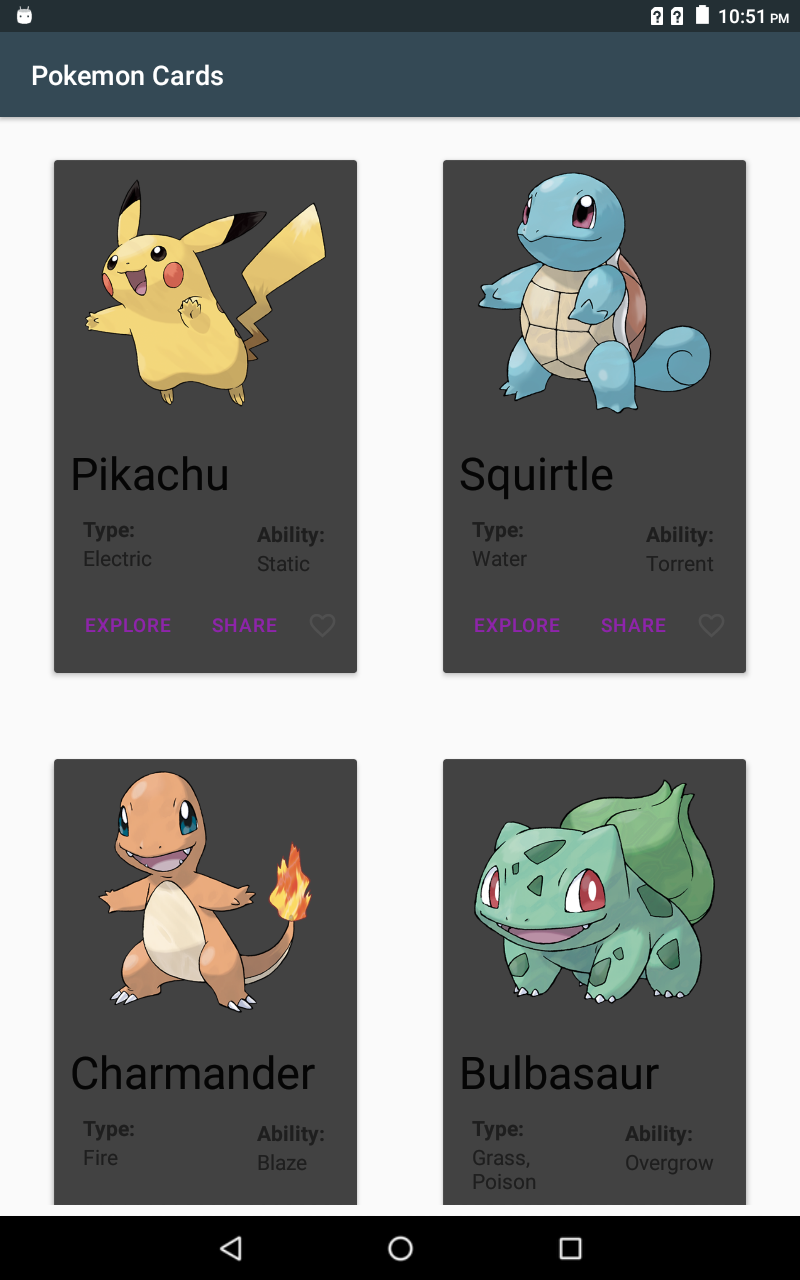
For sides: **maxCardElevation** + (1 - cos45) \* **cornerRadius**

For top and bottom: **maxCardElevation** \* 1.5 + (1 - cos45) \* **cornerRadius**.

Also, child views are not clipped when their corners intersect with the card’s rounded corners. To arrange these features, they implement two attributes; cardPreventCornerOverlap and cardUseCompatPadding which are explained below.

Default values of Card View attributes are:

* **cardCornerRadius**: 2dp
* **cardElevation**: 2dp
* **cardMaxElevation**: 2dp
* **cardUseCompatPadding**: false
* **cardPreventCornerOverlap**: true



CardView.Dark style is applied.

**Advantages and Disadvantages**

It enables easier to implement complex and custom ListView and RecyclerView. The most important advantage comes from the design part. CardView adds automatically an elevation to your card which causes the shadow effect. Also, you can make your card’s corner rounded. Another advantage is that unlike grid and views, cards can be any length. In the list and grid view all cells are at same size but cards can vary in size. Additionally, unlike layouts, Cards that are the same hierarchy level, can have different types of views or contents.

**Requirements for Implementation**

CardView is a part of v7 support libraries. It is established in Android 5.0 (Lollipop) and added in version 25.1.0. To use cardView and Material Design, add the following requirements to the build.gradle file in the project:

dependencies {  
    implementation 'com.android.support:cardview-v7:27.1.1'  
 implementation 'com.android.support: design:28.0.0-alpha1 }

In XML file, cardView is called as android.support.v7.widget.CardView and if the Material Design’s cardView is preferred, android.support.design.card.MaterialCardView should be used.

**Areas of Usage**

CardView is mostly used with RecyclerView or ListView. Each item of those views can use CardView as a container. Also, CardView is used to show heterogeneous content. For homogeneous content, grid or list view are preferred.

**Material Card View**

\*\*\* MaterialCardView is still in progress implementation, new features and updates will come.

MaterialCardView adapts all the features of CardView and add new ones which are ***strokeColor*** and ***strokeBorder***.

In addition to base card view, material card view adds two new features as mentioned above and customizes the elevations and background color. In material design, background color is fixed as white (still you can customize with cardBackgroundColor attribute).

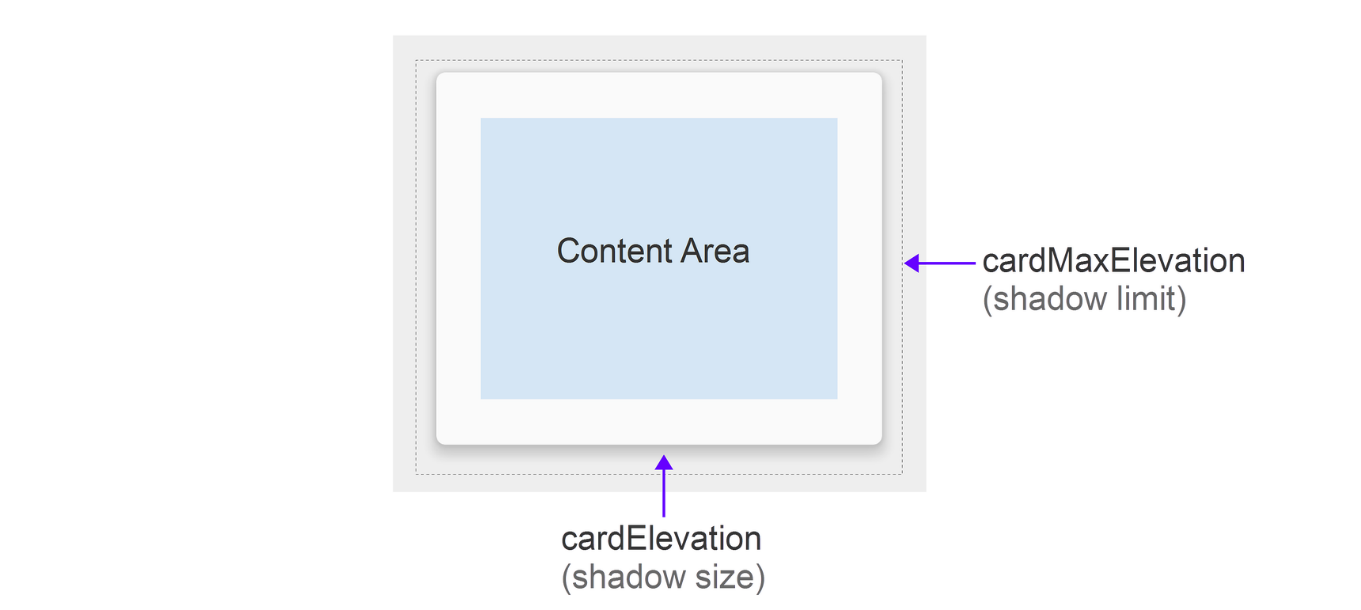
**Style**

Material card view has only one default style which is MaterialCardView. In the style, card ‘s margin cannot be set.

**Attributes**

The list of Material Card View’s Attributes:

* ***cardCornerRadius***: adjust the roundness of the card’s corners
* ***cardElevation***: adjust the elevation (shadow) of the card
* ***cardMaxElevation***: arrange backward well-matched maximum elevation. It becomes useful when the elevation of the card is changed dynamically. This attribute prevents the shadow from crossing the limit that is set by cardMaxElevation attribute.



\*\*\* “Calling this method has no effect if device OS version is Lollipop or newer and cardUseCompatPadding is false.” *Android Developer*

* ***cardUseCompatPadding***: true/false. If the shadow of the card view is overflows parent view of the card, this feature blocks to clipping the shadow of the card by adding extra padding around the card. (Lollipop and after)
* ***cardPreventCornerOverlap***: true/false. Boundaries of the card is not clipped for the rounded corners on previous platforms from Lollipop. To avoid overlapping with rounded corners, it adds padding to content. To disable this feature, set this attribute to *false.*
* ***cardBackgroundColor***: adjust background color of the card
* ***strokeColor***: set the stroke color of the card
* ***strokeWidth***: change the thickness of the stroke of the card

Not use in style tag:

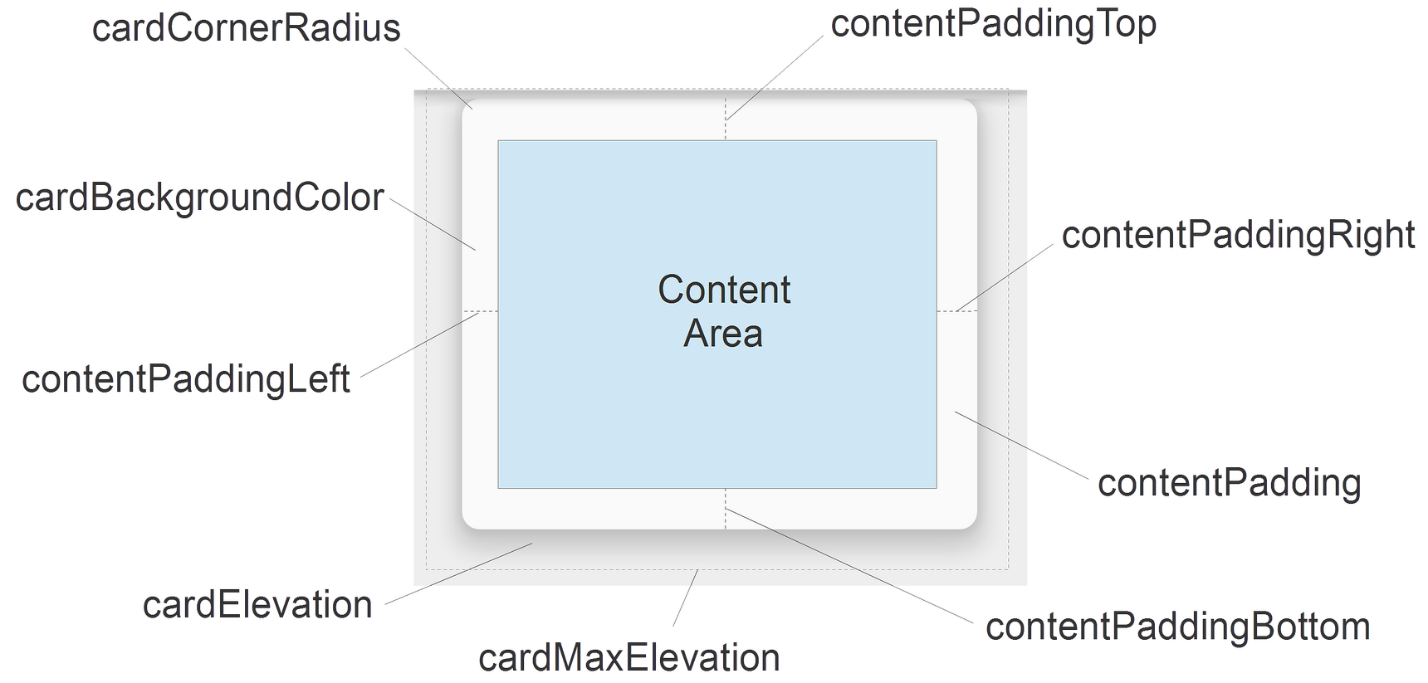
* ***contentPadding*** and ***contentPaddingBottom/Top/Left/Right***: add padding between the edges and child views.

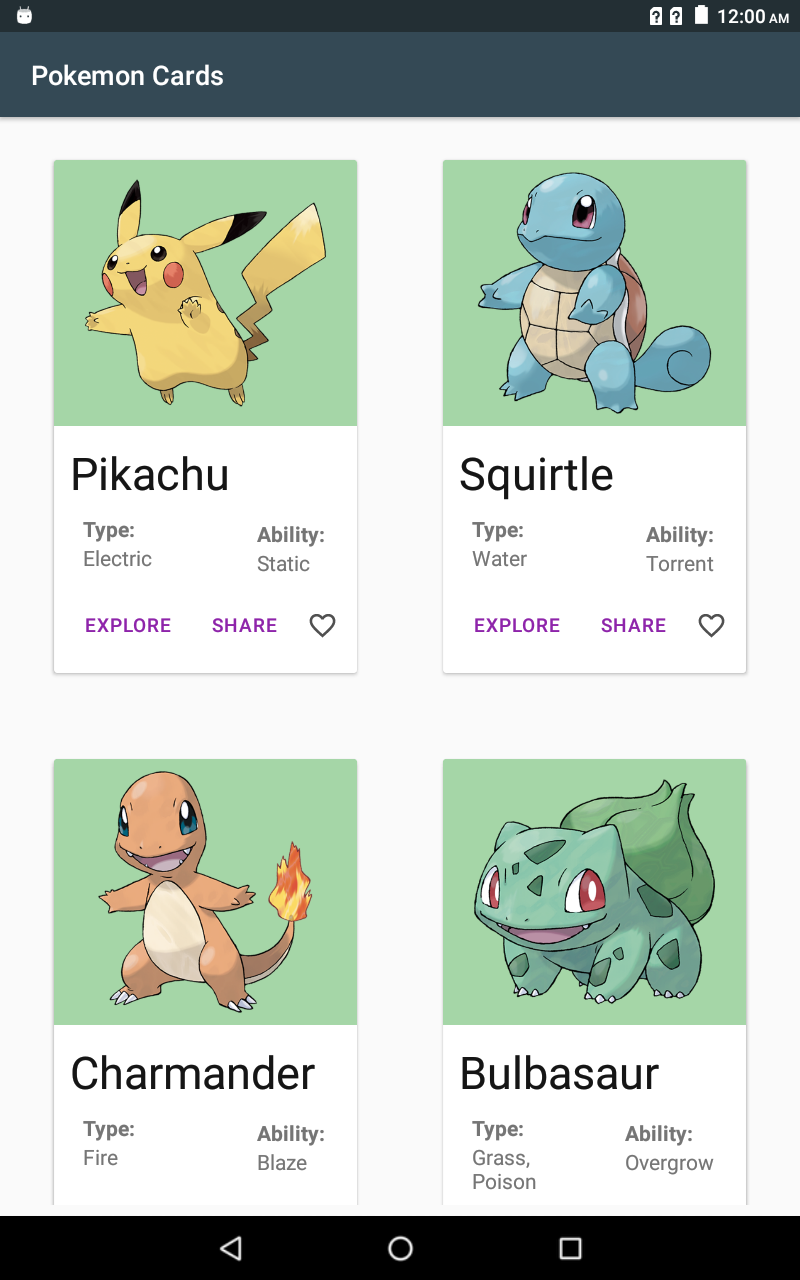
\***Stroke color** and **stroke width** are not written as *android:strokeColor* and *android:strokeWidth* in style tag.

\*Shadow of the card is clipped if it exceeds the parent view’s borders or padding space. There are two options to avert this; one is adding clipToPadding = false to the parent view and adding extra padding. Other one is using cardUseCompatPadding = true statement in the card.

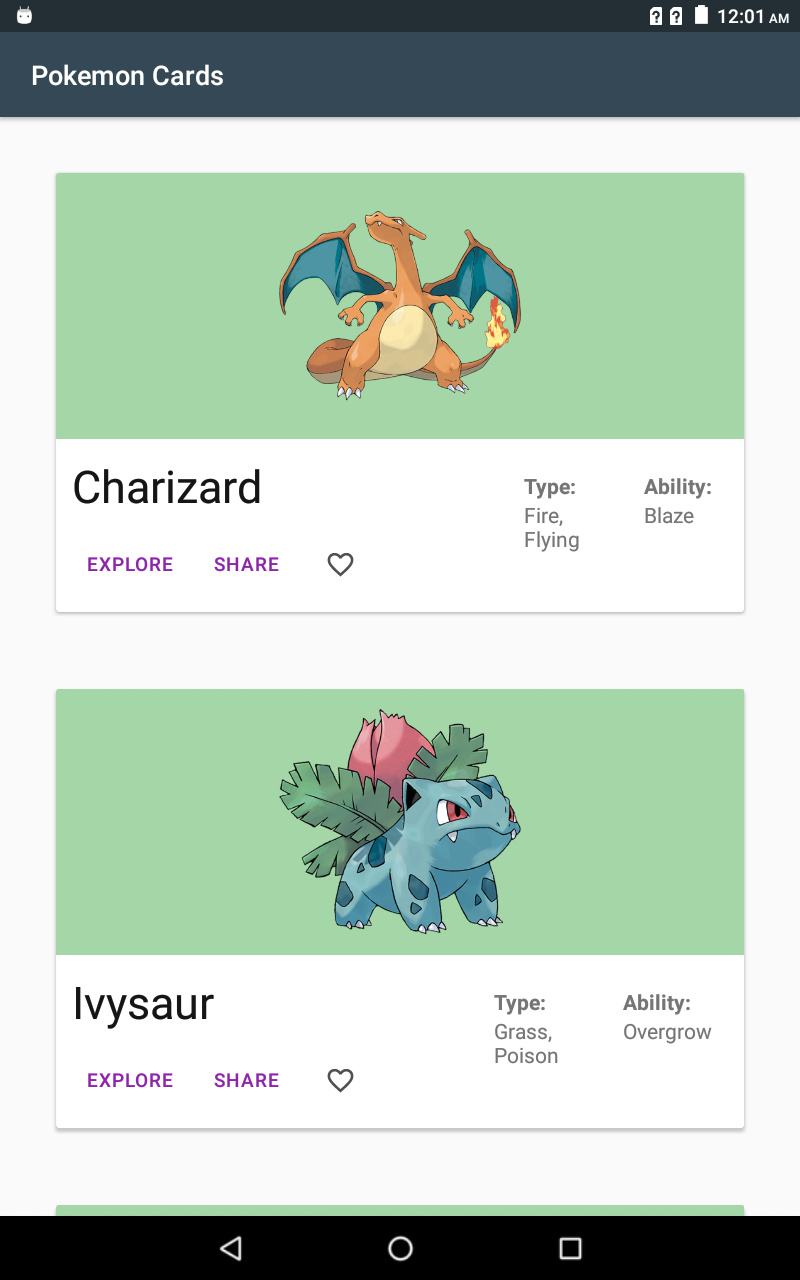
Values of Material Card View’s attributes are:

* **cardElevation**: 1dp
* **cardBackgroundColor**: #ffffff (white)

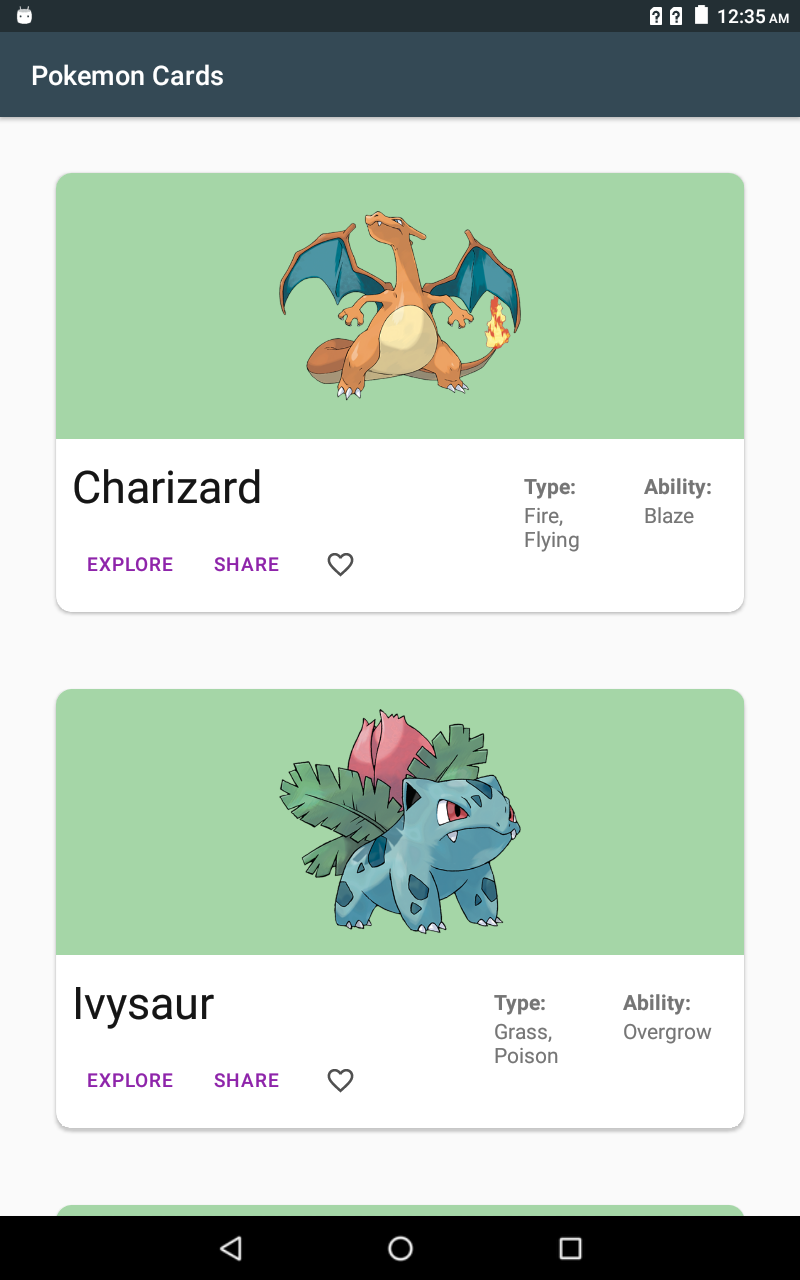




Default Material Pokemon Card View in GridLayout



Default Material Pokemon Card View in ListView

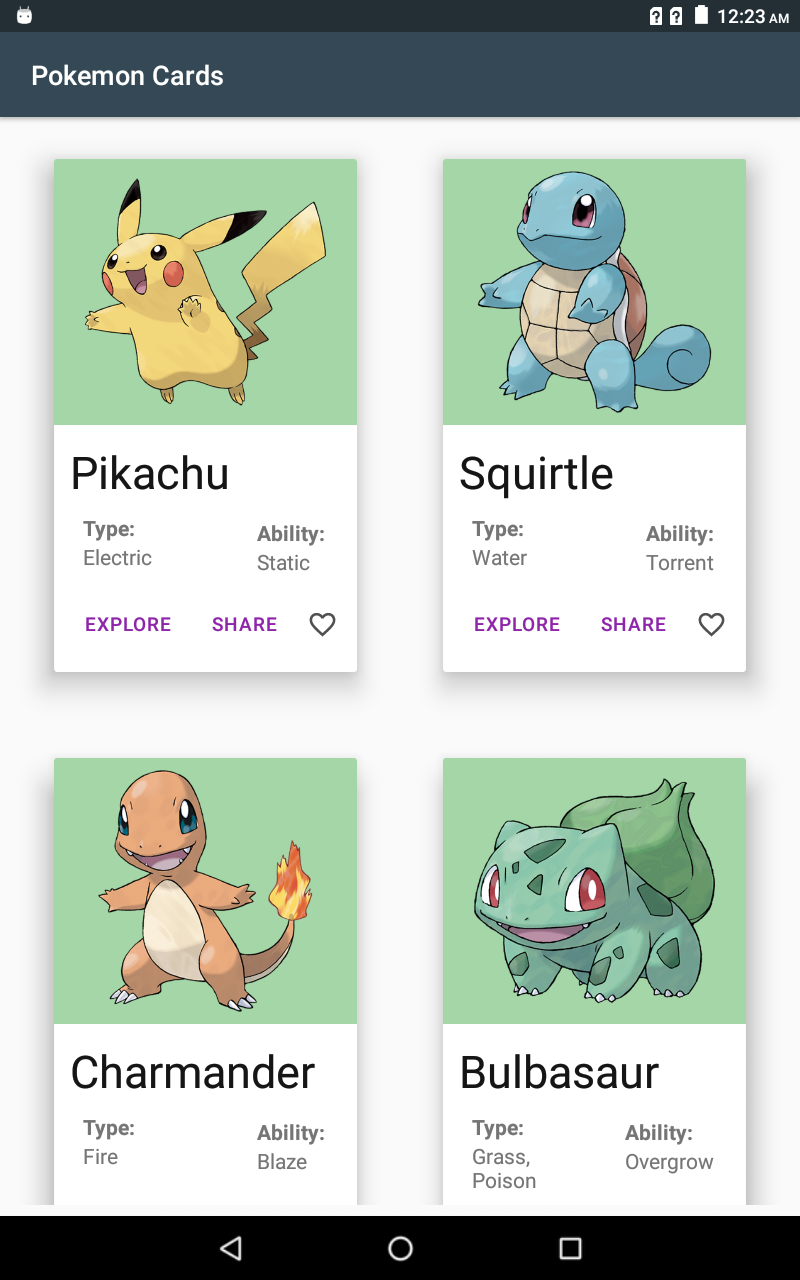


<style name="cardCornerRadius" parent="Widget.MaterialComponents.CardView">

<item name="cardCornerRadius">12dp</item>

</style>

Customize the corner radius of the card: 12dp

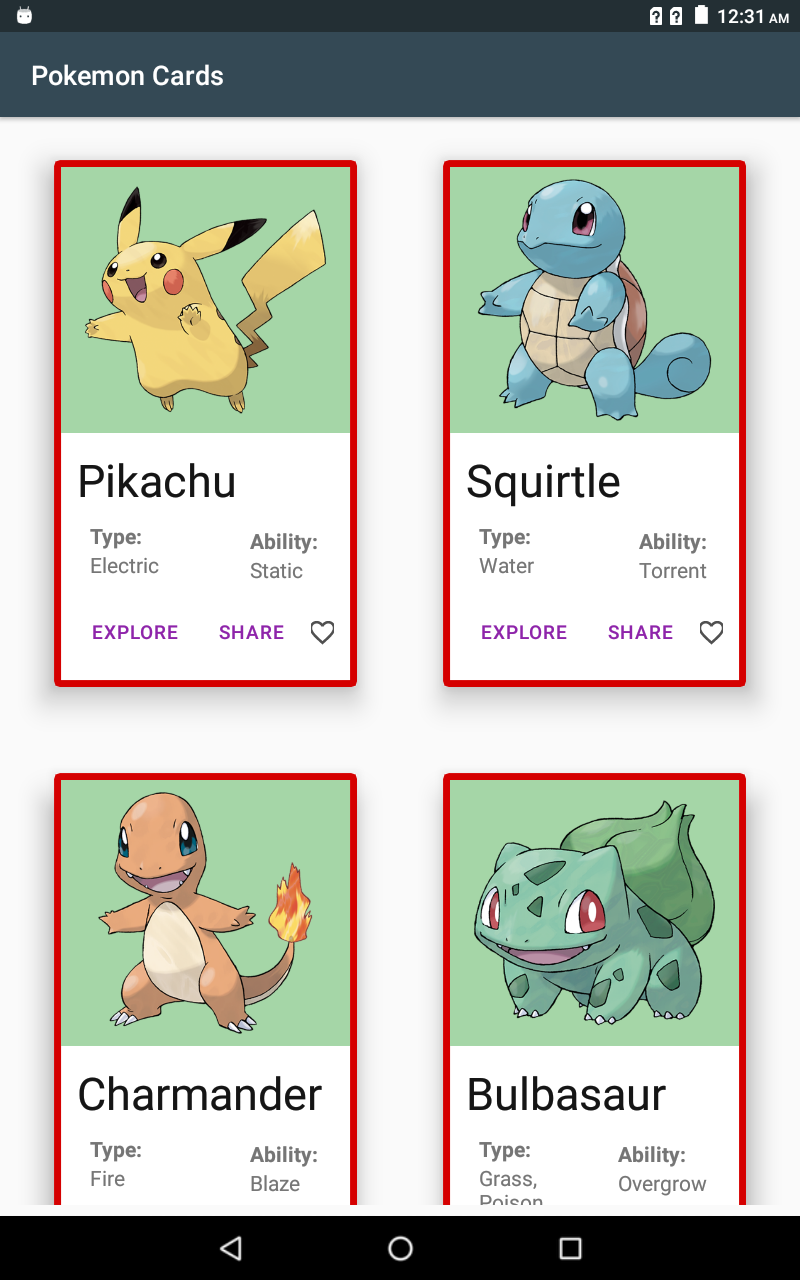


<style name="cardElevation" parent="Widget.MaterialComponents.CardView">

<item name="cardElevation">15dp</item>

<item name="cardMaxElevation">30dp</item>

</style>



<style name="cardElevation" parent="Widget.MaterialComponents.CardView">

<item name="strokeColor">#D50000</item>

<item name="strokeWidth">5dp</item>

</style>

Customize the stroke color and width of the cards:

strokeColor: #d50000, strokeWidth: 5dp

**References**

Android Developers. (n.d.). Retrieved June 27, 2018, from https://developer.android.com/

Homepage. (n.d.). Retrieved June 27, 2018, from <https://material.io/>

M. (n.d.). CardView - Xamarin. Retrieved from https://docs.microsoft.com/en-us/xamarin/android/user-interface/controls/card-view