ADS
DR2 Des. Android

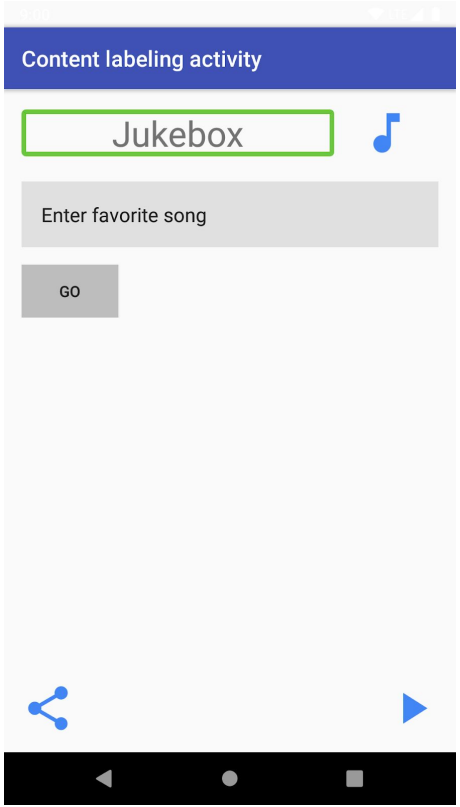
Desenvolvimento de Interfaces Android

Análise e Desenvolvimento de Software - 2020 / 4

Rotulagem de Conteúdo

Quando um usuário com deficiência visual tenta navegar em um aplicativo pelo toque, o TalkBack anuncia todo o conteúdo acionável, desde que tenha rótulos significativos, úteis e descritivos.

Se esses rótulos estiverem ausentes, o TalkBack pode não ser capaz de explicar adequadamente a função de um controle específico para um usuário. Em alguns casos, o TalkBack pode pular completamente algum conteúdo.



> Dica: é uma prática recomendada usar **android: hint em** vez de **contentDescription** em **EditTexts** .

Rotulagem de Conteúdo

```xml
<ImageView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:contentDescription="@null"
    android:src="@drawable/ic_music_note" />
```

```xml
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentStart="true"
    android:contentDescription="@string/share"
    android:background="@null"
    android:src="@drawable/ic_share" />
```

Rotulagem de Conteúdo

```
private void updateImageButton() {
    if (mPlaying) {
        mPlayPauseToggleImageView.setImageResource(R.drawable.ic_pause);
    } else {
        mPlayPauseToggleImageView.setImageResource(R.drawable.ic_play_arrow);
    }
}
```

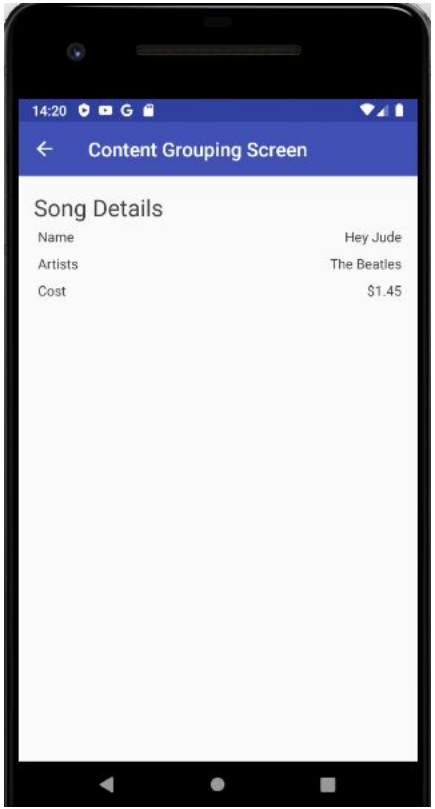
```
private void updateImageButton() {
    if (mPlaying) {
        mPlayPauseToggleImageView.setImageResource(R.drawable.ic_pause);
        mPlayPauseToggleImageView.setContentDescription(getString(R.string.pause));

    } else {
        mPlayPauseToggleImageView.setImageResource(R.drawable.ic_play_arrow);
        mPlayPauseToggleImageView.setContentDescription(getString(R.string.play));
    }
}
```

Agrupando Conteúdo

Às vezes, o feedback auditivo que o TalkBack fornece para elementos visuais em um aplicativo pode não refletir sua estrutura lógica e espacial.

Mesmo que os elementos possam ser ordenados visualmente de uma maneira sensata, eles podem ser falados fora de ordem.

Embora o usuário do TalkBack seja capaz de descobrir todo o conteúdo na tela, ele precisa deslizar várias vezes. Se os detalhes da música tivessem muito mais campos, a experiência rapidamente se tornaria tediosa.



Agrupando Conteúdo

Como os dados da música são compostos de apenas seis strings curtas, poderíamos fazer o TalkBack agrupar todos os seis itens em um único anúncio. Vamos experimentar essa abordagem.

```
<RelativeLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:focusable="true">
...
</RelativeLayout>
```

Agrupando Conteúdo

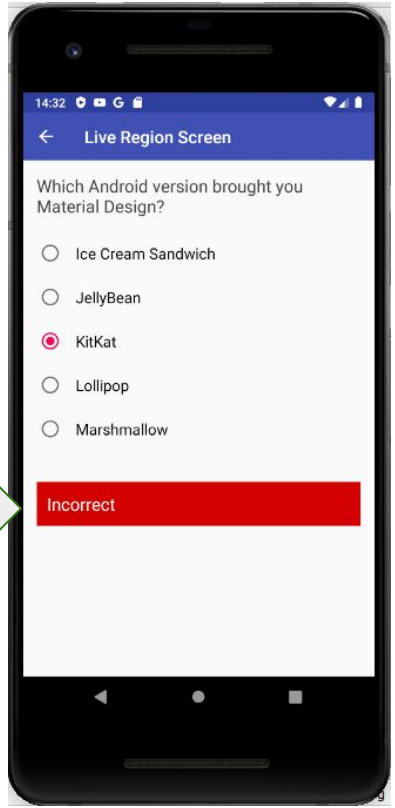
No exemplo que estamos considerando, ter um único anúncio TalkBack é melhor do que seis; mas esta solução tem seus limites:

1. Ele rapidamente se torna não escalável se o número de campos de detalhes da música aumentar ou se os campos contiverem muito texto.
2. Nem a versão original de seis anúncios nem a versão de anúncio único que acabamos de implementar levam em consideração o agrupamento natural de visualizações. Os dados são claramente organizados visualmente em duas colunas - os itens à esquerda e os valores correspondentes à direita. Esse agrupamento é evidente para um usuário que pode ver a tela, mas você deve garantir que isso seja igualmente óbvio para um usuário com deficiência visual com o TalkBack.

Usando uma Live Region

Até agora, todos os exemplos envolveram o uso do Talkback com visualizações nas quais o usuário se concentrou explicitamente. No entanto, às vezes você precisa descobrir um texto que é atualizado dinamicamente sem ter que navegar explicitamente até ele e se concentrar nele.

```
<TextView
    android:id="@+id/feedback_text_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@null"
    android:textColor="@color/white"
    android:padding="@dimen/standard_padding"
    android:textSize="@dimen/large_text"
    android:accessibilityLiveRegion="polite" />
```



Usando uma Live Region

Atribuímos ao nosso **accessibilityLiveRegion** o valor de "polite". Isso simplesmente significa que o TalkBack não interromperá nada que já esteja anunciando ao processar o evento gerado por uma região ao vivo.

Uma alternativa para "polite" é "assertive", que instrui o TalkBack a mover anúncios relacionados a uma região ao vivo na fila de eventos e interromper os anúncios em andamento.

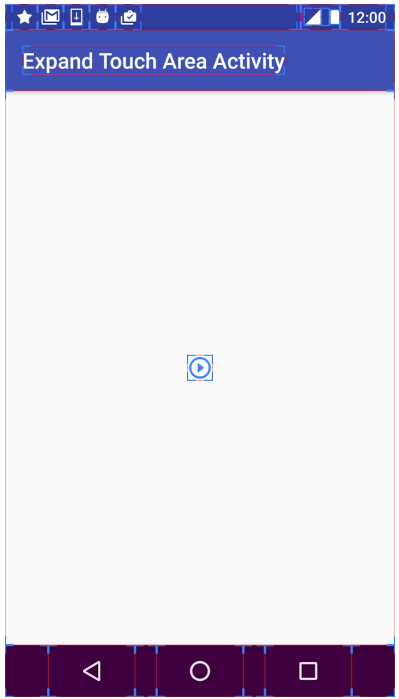
Você deve usar regiões ativas com moderação e quase sempre evitar regiões ativas "assertivas".

Áreas de Toque Acessíveis

Muitas pessoas têm dificuldade em se concentrar em pequenas áreas de toque na tela.

Isso pode ocorrer porque seus dedos são grandes ou porque eles têm uma condição médica que prejudica suas habilidades motoras.

Pequenas áreas de toque também tornam mais difícil para os usuários de leitores de tela navegar pelos aplicativos usando “explorar pelo toque”.




```
<ImageButton
    ...
    android:minWidth="48dp"
    android:minHeight="48dp" />
```

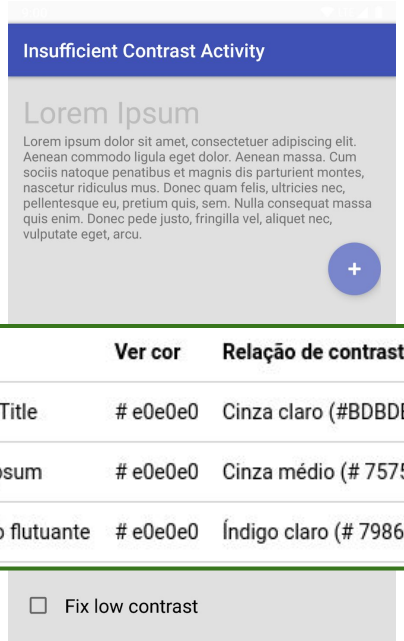
Contraste de Cor Adequado

Usuários com baixa visão não podem ler informações em uma tela se não houver contraste suficiente entre o **primeiro plano e o fundo**.

As taxas de contraste baixas entre as cores do primeiro plano e do plano de fundo podem fazer com que as visualizações fiquem desfocadas para alguns usuários, enquanto as taxas de contraste altas as tornam mais claras.

Diferentes situações de iluminação podem amplificar os problemas criados por baixas taxas de contraste.

Contraste de Cor Adequado



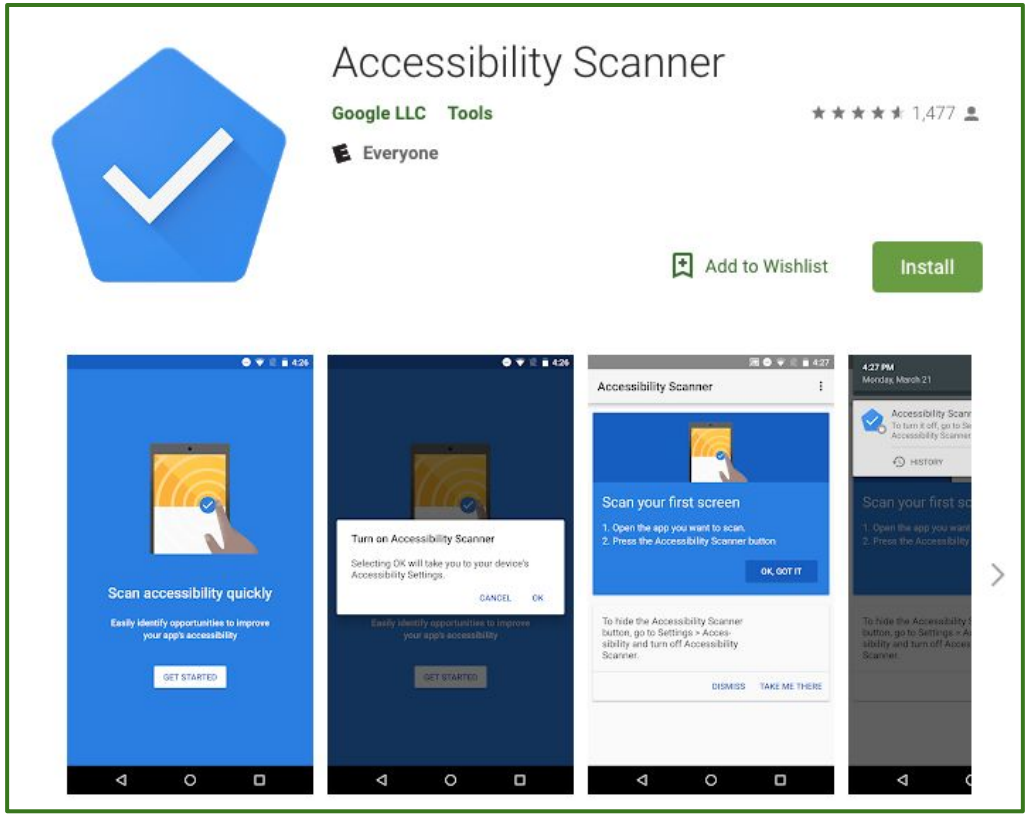
fundo	Ver cor	Relação de contraste	
Lorem ipsum Title	# e0e0e0	Cinza claro (#BDBDBD)	**1.42:1**
Texto lorem ipsum	# e0e0e0	Cinza médio (# 757575)	**3.49:1**
Botão de ação flutuante	# e0e0e0	Índigo claro (# 7986CB)	**2.61:1**

fundo	Ver cor	Relação de contraste	
Lorem ipsum Title	#FFFFFF	Cinza claro (# 757575)	**4.61:1**
Texto lorem ipsum	#FFFFFF	Cinza escuro (# 424242)	**10.05:1**
Botão de ação flutuante	#FFFFFF	Indigo (# 303F9F)	**8.98:1**

As Diretrizes de acessibilidade de conteúdo da Web recomendam uma taxa de contraste mínima de **4,5: 1** para todo o texto, com uma taxa de contraste de **3,0: 1** considerada aceitável para texto grande ou em negrito, e você deve tentar atender ou exceder essas taxas de contraste em seus aplicativos.

Scanner de Acessibilidade

O Google oferece uma ferramenta de acessibilidade que sugere melhorias para aplicativos Android - como aumentar pequenos alvos de toque, aumentar o contraste e fornecer descrições de conteúdo - para que pessoas com necessidades de acessibilidade possam usar seu aplicativo com mais facilidade.



Laboratório de Acessibilidade 2

Esse é um outro Code Lab (atualização?) que complementa o anterior.

O foco principal deste laboratório é o Scanner de Acessibilidade.

https://codelabs.developers.google.com/codelabs/starting-android-accessibility

Exercício

1. Instalar o **Accessibility Scanner**.
2. Explorar o CodeLabs de Acessibilidade.
3. Passar o Accessibility Scanner nas suas calculadoras do TP1.