ANDROID ACCESSIBILITY

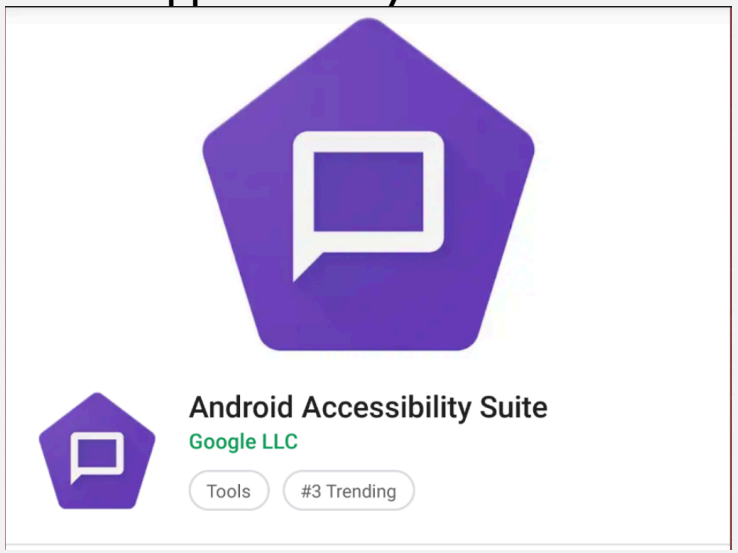
My Spectrum App – 11/05/2018

-Saamer Mansoor @saamerm

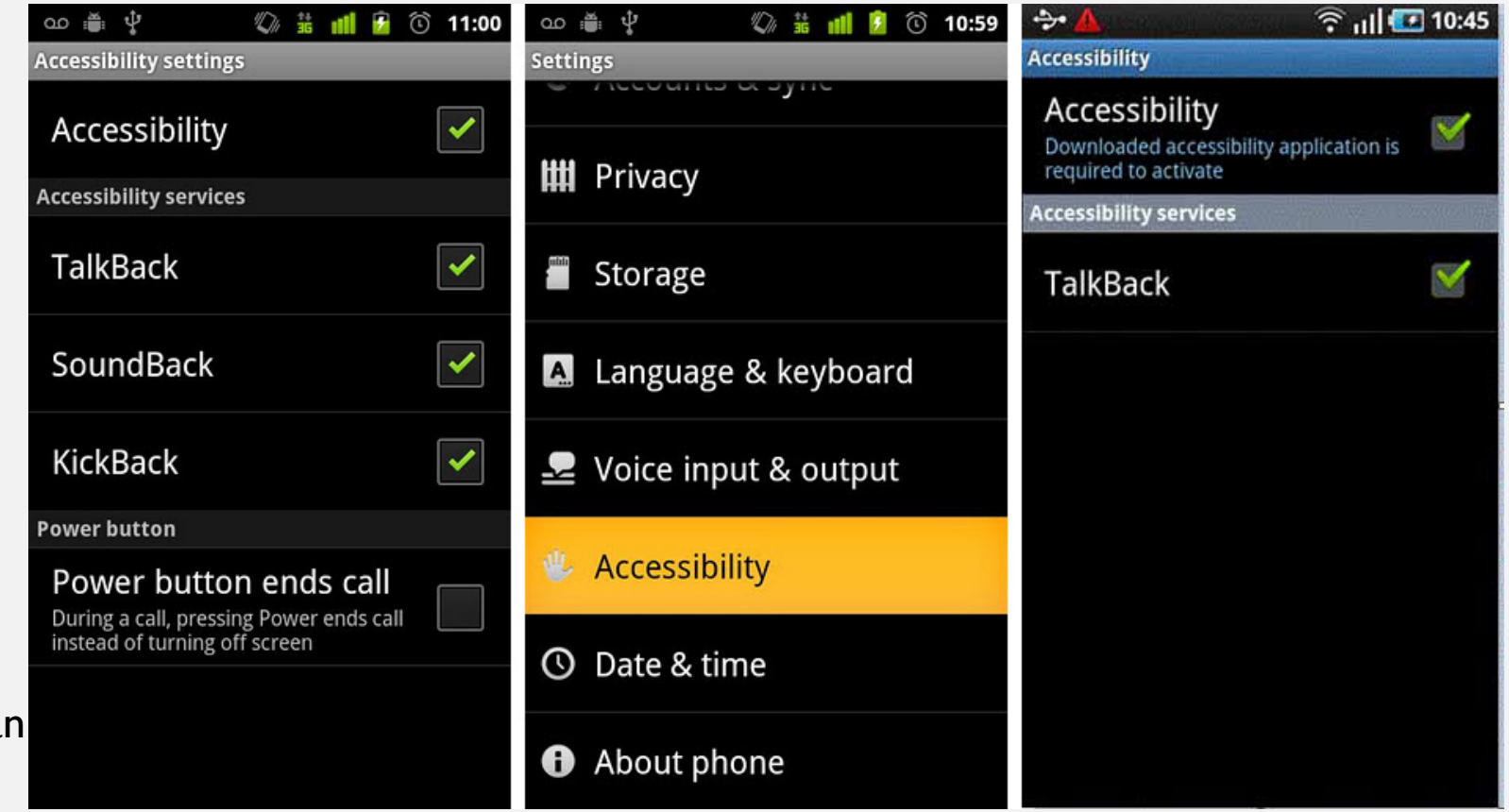
TALKBACK ACTIVATION

Rebranded as “Android Accessibility Suite” on June 21, 2018

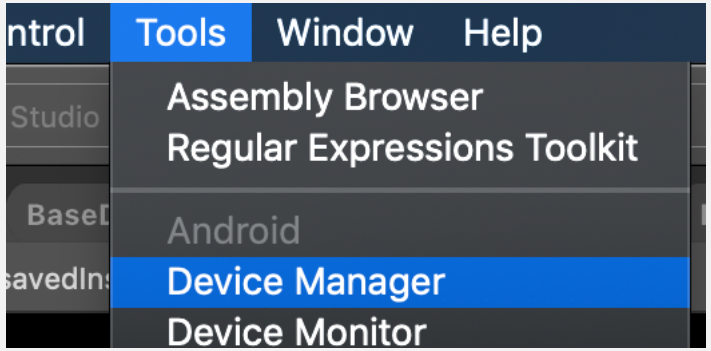
Download “Android Accessibility Suite” app from Play Store:

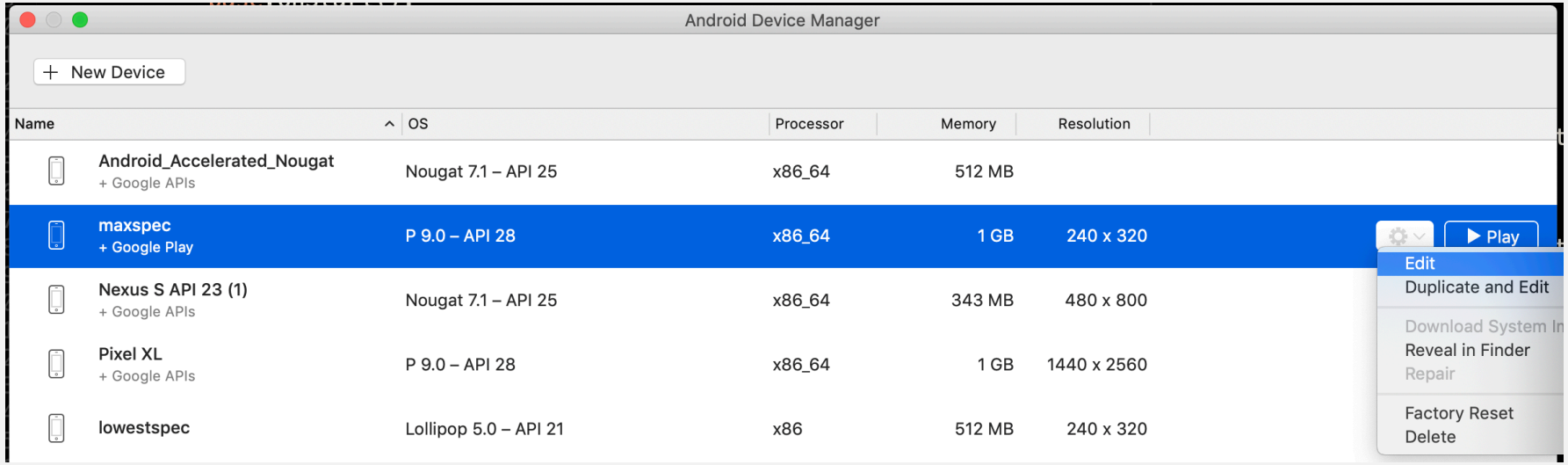


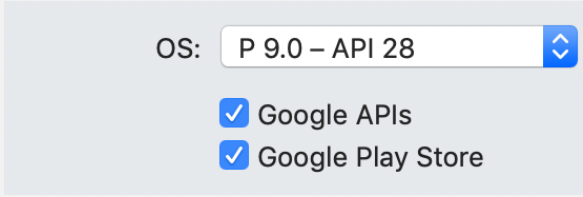
Then go to Settings and Turn Talkback On. Note that it may look different than what you see on the right->

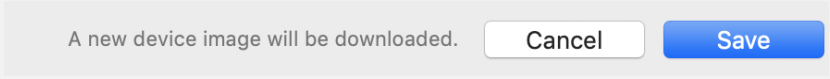


TALKBACK ON EMULATOR

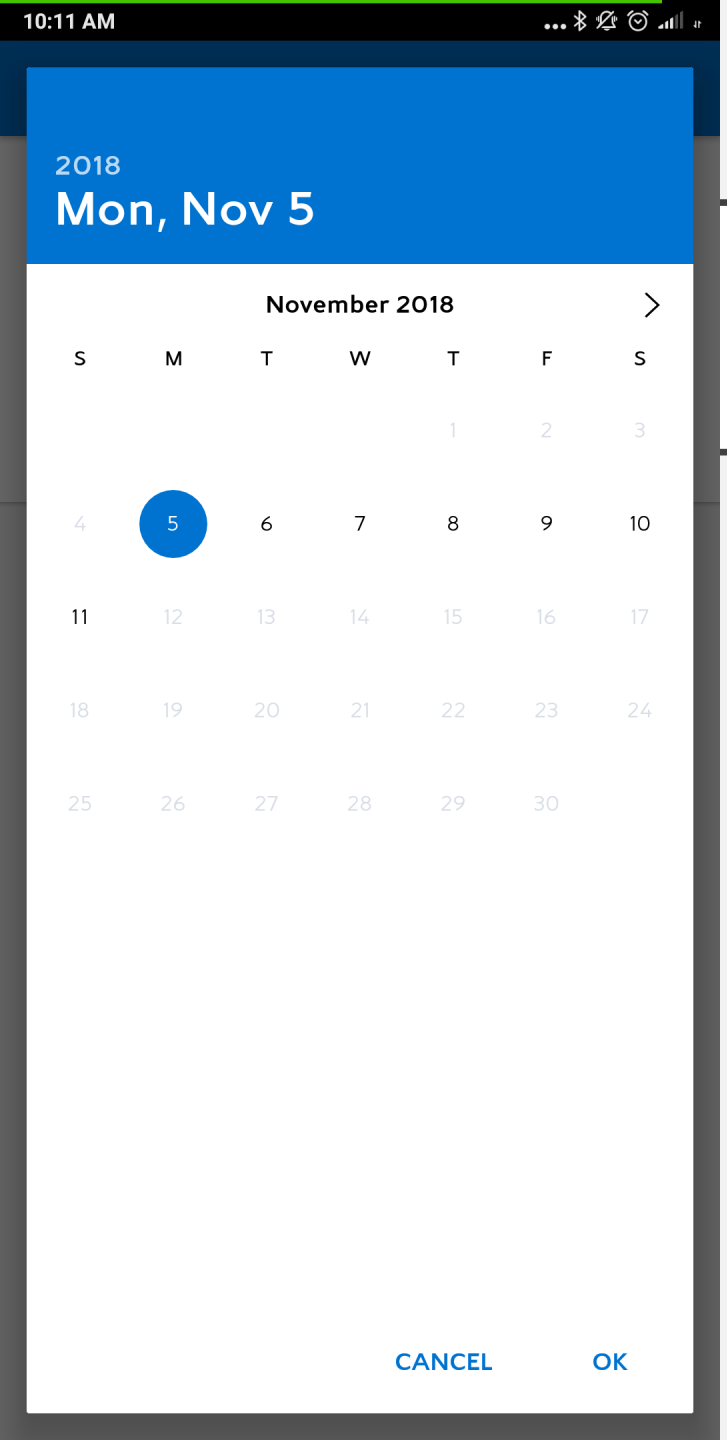








And then follow instructions from the previous slide, to download Talkback from the Android Play Store on the Emulator device.



BASIC GESTURES

Action	Gesture
Move to next item on screen	Swipe right
Move to previous item on screen	Swipe left
Cycle through navigation settings	Swipe up or down
Select focused item	Double-tap

BACK & FORTH GESTURES

Action	Swipe
Move to first item on screen	Up then down
Move to last item on screen	Down then up
Scroll forward (if you're on a page longer than one screen)	Right then left
Scroll back (if you're on a page longer than one screen)	Left then right
Move slider up (such as volume)	Right then left
Move slider down (such as volume)	Left then right

(RIGHT) ANGLE GESTURES

These gestures are two-part swipes at a right angle. For example, the default gesture for going to the Home screen is to swipe up then left at a sharp 90-degree angle.

Action	Swipe
Home button	Up then left
Back button	Down then left
Overview button	Left then up
Notifications	Right then down (see note below)
Open local context menu	Up then right
Open global context menu	Down then right

Note: The Notifications gesture (right then down) is available only in some TalkBack versions. If this gesture doesn't work for you, use a two-finger swipe down from the top of the screen to open the notifications shade.

FOCUS AND ANNOUNCE

Accessibility Utility class to be used:

```
ng Android.Views.Accessibility;
ng MobileCareAppAndroid.DependencyInjection;

espace MobileCareAppAndroid.Util

public static class Accessibility
{
    /// <summary> Announce a message if accessibility is turned on. Primary use woul ...
    public static void Announce(string message)
    {
        var manager = AccessibilityManager.FromContext(Presenter.ActivityTracker.CurrentActivity);
        if (manager.IsEnabled)
        {
            var accessEvent = AccessibilityEvent.Obtain(EventTypes.Announcement);
            accessEvent.Text.Add(new Java.Lang.String(message));
            manager.SendAccessibilityEvent(accessEvent);
        }
    }

    /// <summary> Focuses on the passed in view Warning: If you want to do this comi ...
    public static void Focus(View viewToAnnounce)
    {
        if (AccessibilityManager.FromContext(Presenter.ActivityTracker.CurrentActivity).IsEnabled)
            viewToAnnounce.SendAccessibilityEvent(EventTypes.ViewFocused);
    }
}
```

FOCUS AND ANNOUNCE

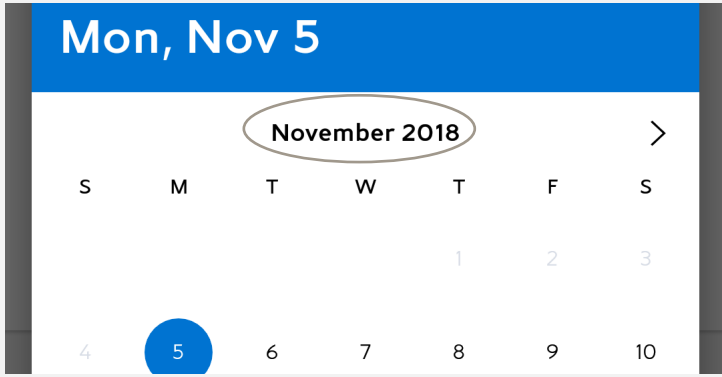
BaseDialogFragment implementation uses the Accessibility Utility class:

```
/// <summary> Focuses on the passed in view Warning: If you want to do this comi ...
public void AccessibilityFocus(View viewToAnnounce)
{
    Accessibility.Focus(viewToAnnounce);
}

/// <summary> Announce a message if accessibility is turned on. Primary use woul ...
public void AccessibilityAnnounce(string message)
{
    Accessibility.Announce(message);
}
```

CalendarDialogFragment's OnCreateView ">" button:

```
// right arrow to move to next month
_nextMonth = rootView.FindViewById<ImageView>(Resource.Id.next);
_nextMonth.Click += (sender, e) =>
{
    _pager.CurrentItem = _pager.CurrentItem + 1;
    base.AccessibilityAnnounce(currentMonth.Text);
    base.AccessibilityFocus(currentMonth);
};
```



Now on moving to the next month, talkback focuses and announces November 2018

CONTENT DESCRIPTION

```
// Tests show the image buttons as unlabeled
_prevMonth.ContentDescription = AppResources.AccessibilityCalendarPreviousMonthIcon;
_nextMonth.ContentDescription = AppResources.AccessibilityCalendarNextMonthIcon;
```

Done inside OnCreateView() Now, focusing on the next button doesn't say "Unlabeled button", it says "Next Month, button"

For non-Android devs, you can also edit attributes in the axml layout:

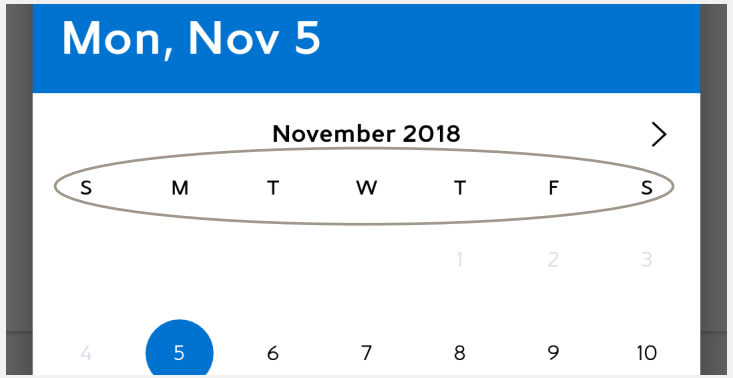
```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:contentDescription="@string/share"
    android:src="@drawable/ic_share"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

REQUIRED FOR ACCESSIBILITY

Focusability and FocusedByDefault didn't seem to do anything useful for talkback.
Hint attribute is to tell the user the type of control, eg: button. "Next Month, button"

CalendarDialogFragment.axml layout:

```
<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:gravity="center"
    style="@style/P4.Medium"
    android:importantForAccessibility="no"
    android:text="S" />
<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:gravity="center"
    style="@style/P4.Medium"
    android:importantForAccessibility="no"
    android:text="M" />
```



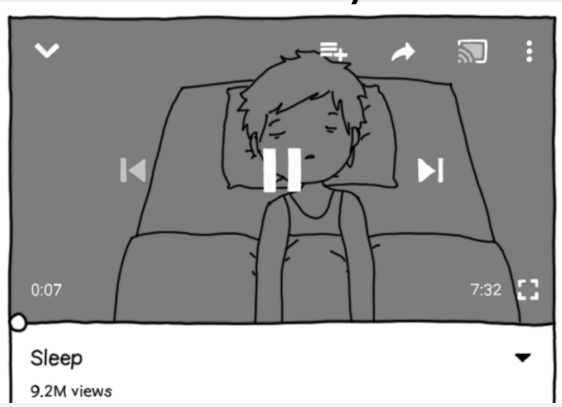
Now, when you scroll after the ">" button, it skips all the days of the month.
As shown, in the previous slide, you can edit the attributes of the controls in C# as well

ACCESSIBILITY MANAGER

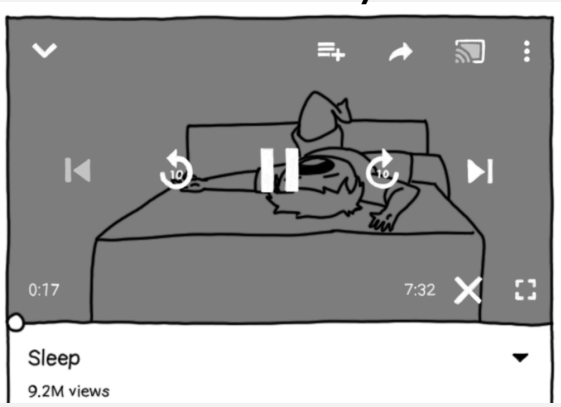
Checks the context for accessibility information, and implement accessibility specific UI behavior, within OnCreateView()

```
// Hides the top two rows of the calendar if Accessibility TalkBack is turned On
// This causes the currentMonth to be announced first on PageLoad
var manager = AccessibilityManager.FromContext(Activity);
_dateText.Visibility = manager.IsTouchExplorationEnabled ? ViewStates.Gone : ViewStates.Visible;
_dateYear.Visibility = manager.IsTouchExplorationEnabled ? ViewStates.Gone : ViewStates.Visible;
```

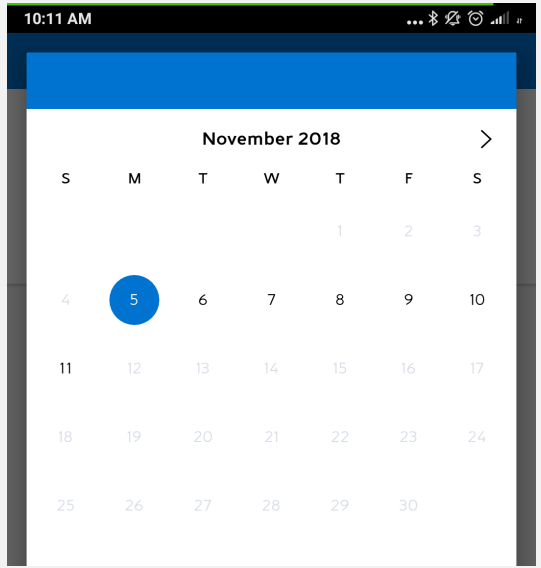
Accessibility Off



Accessibility On



<-YouTube app example of different experiences for accessibility users



ACCESSIBILITY DELEGATE

Override the existing AccessibilityDelegate of a button/image/label control within the OnCreateView() :

```
}
NextMonthButtonAccessibilityDelegate accessDelegate = new NextMonthButtonAccessibilityDelegate();
_nextMonth.SetAccessibilityDelegate(accessDelegate);
```

```
public class NextMonthButtonAccessibilityDelegate : View.AccessibilityDelegate
{
    private bool _isFirstPageLoad = true;

    public override void OnInitializeAccessibilityEvent(View host, AccessibilityEvent e)...

    public override void OnPopulateAccessibilityEvent(View host, AccessibilityEvent e)...

    //Works to prevent announcement of accessibility on first Instance
    public override void SendAccessibilityEvent(View host, [GeneratedEnum] EventTypes eventType)
    {
        if (_isFirstPageLoad && host.GetType() == typeof(ImageView))
        {
            _isFirstPageLoad = false;
            ImageView imageView = (ImageView)host;
            if (imageView.Id != Resource.Id.next)
            {
                base.SendAccessibilityEvent(host, eventType);
            }
        }
        else
            base.SendAccessibilityEvent(host, eventType);
    }

    //Works to prevent focus, RequiredForAccessibility = no for the first time
    public override void OnInitializeAccessibilityNodeInfo(View host, AccessibilityNodeInfo info)...
}
```

Add the AccessibilityDelegate within the same CalendarDialogFragment class at the bottom.

Overriding each of those, can be useful in a certain way, SendAccessibilityEvent() and OnInitializeAccessibilityNodeInfo() were useful for our testing as you can see the