Building For TV Devices

TV devices support has been implemented with the intention of making existing React Native applications work on Apple TV and Android TV, with few or no changes needed in the JavaScript code for the applications.

Android TV 🚧 tvOS

Build changes

- *Native layer*: To run React Native project on Android TV make sure to make the following changes to `AndroidManifest.xml`

```
 <!-- Add custom banner image to display as Android TV launcher icon -->
<application
 ...
 android:banner="@drawable/tv_banner"
 >
   ...
   <intent-filter>
     ...
     <!-- Needed to properly create a launch intent when running on Android TV -->
     <category android:name="android.intent.category.LEANBACK_LAUNCHER"/>
   </intent-filter>
   ...
 </application>
```

- *JavaScript layer*: Support for Android TV has been added to `Platform.android.js`. You can check whether code is running on Android TV by doing

```
const Platform = require('Platform');
const running_on_android_tv = Platform.isTV;
```

Code changes

- *Access to touchable controls*: When running on Android TV the Android framework will automatically apply a directional navigation scheme based on relative position of focusable elements in your views. The `Touchable` mixin has code added to detect focus changes and use existing methods to style the components properly and initiate the proper actions when the view is selected using the TV remote, so `TouchableWithoutFeedback`, `TouchableHighlight`, `TouchableOpacity` and `TouchableNativeFeedback` will work as expected. In particular:
  - `onFocus` will be executed when the touchable view goes into focus
  - `onBlur` will be executed when the touchable view goes out of focus
  - `onPress` will be executed when the touchable view is actually selected by pressing the "select" button on the TV remote.
- *TV remote/keyboard input*: A new native class, `ReactAndroidTVRootViewHelper`, sets up key events handlers for TV remote events. When TV remote events occur, this class fires a JS event. This event will be picked up by instances of the `TVEventHandler` JavaScript object. Application code that needs to implement custom handling of TV remote events can create an instance of `TVEventHandler` and listen for these events, as in the following code:

```
const TVEventHandler = require('TVEventHandler');

class Game2048 extends React.Component {
  _tvEventHandler: any;

  _enableTVEventHandler() {
    this._tvEventHandler = new TVEventHandler();
    this._tvEventHandler.enable(this, function (cmp, evt) {
      if (evt && evt.eventType === 'right') {
        cmp.setState({board: cmp.state.board.move(2)});
      } else if (evt && evt.eventType === 'up') {
        cmp.setState({board: cmp.state.board.move(1)});
      } else if (evt && evt.eventType === 'left') {
        cmp.setState({board: cmp.state.board.move(0)});
      } else if (evt && evt.eventType === 'down') {
        cmp.setState({board: cmp.state.board.move(3)});
      } else if (evt && evt.eventType === 'playPause') {
        cmp.restartGame();
      }
    });
  }

  _disableTVEventHandler() {
    if (this._tvEventHandler) {
      this._tvEventHandler.disable();
      delete this._tvEventHandler;
    }
  }

  componentDidMount() {
    this._enableTVEventHandler();
  }

  componentWillUnmount() {
    this._disableTVEventHandler();
  }
}
```

- *Dev Menu support*: On the emulator, cmd-M will bring up the Dev Menu, similar to Android. To bring it up on a real Android TV device, press the menu button or long press the fast-forward button on the remote. (Please do not shake the Android TV device, that will not work :))

- *Known issues*:

  - `TextInput` components do not work for now (i.e. they cannot receive focus automatically, see this comment).
    - It is however possible to use a ref to manually trigger `inputRef.current.focus()`.
    - You can wrap your input inside a `TouchableWithoutFeedback` component and trigger focus in the `onFocus` event of that touchable. This enables opening the keyboard via the arrow keys.
    - The keyboard might reset its state after each keypress (this might only happen inside the Android TV emulator).
  - The content of `Modal` components cannot receive focus, see this issue for details.

Is this page useful? 👍 👎

Edit this page

*Last updated on **Jun 21, 2023***