Native Mobile Apps in JavaScript

Using Exponent and React Native

Charlie Cheever

Exponent

CS50 Seminar
October 28, 2016

About Me

Harvard

Amazon

Facebook

Quora

Exponent

A Brief History of Mobile Development

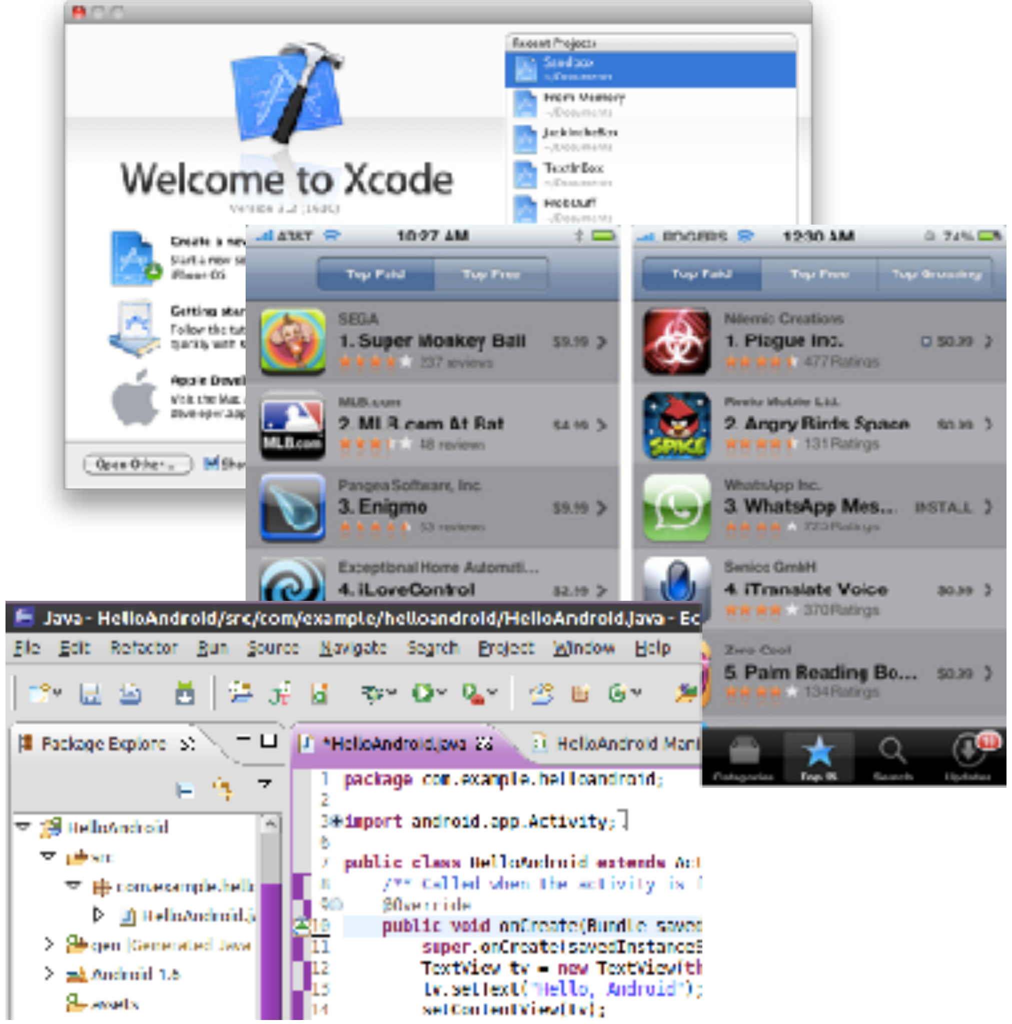
Mobile Websites

- 2007
- Original iPhone had no App Store or any way to build apps without jailbreaking
- People could build mobile websites in Safari and save shortcuts to their home screen.
- No one ever did this

3G
9:42 AM
The New York Times - Breaking News, World...
www.nytimes.com/
Google
The New York Times
THE INAUGURATION
PRESIDENT OBAMA VOWS ERA OF RESP

Native Apps

- 2008
- Apple releases a way to use Xcode to write apps in Objective-C for the iPhone
- Distributed through the App Store
- Google launches Android and allows developers to write apps in Java



```
package com.example.helloandroid;

import android.app.Activity;

public class HelloAndroid extends Act
    /** Called when the activity is f
    @Override
    public void onCreate(Bundle saved
        super.onCreate(savedInstanceS
        TextView tv = new TextView(th
        tv.setText("Hello, Android");
        setContentView(tv);
```

Challenges with Developing Native Apps

- Compile times get bigger and bigger the more you add to your app

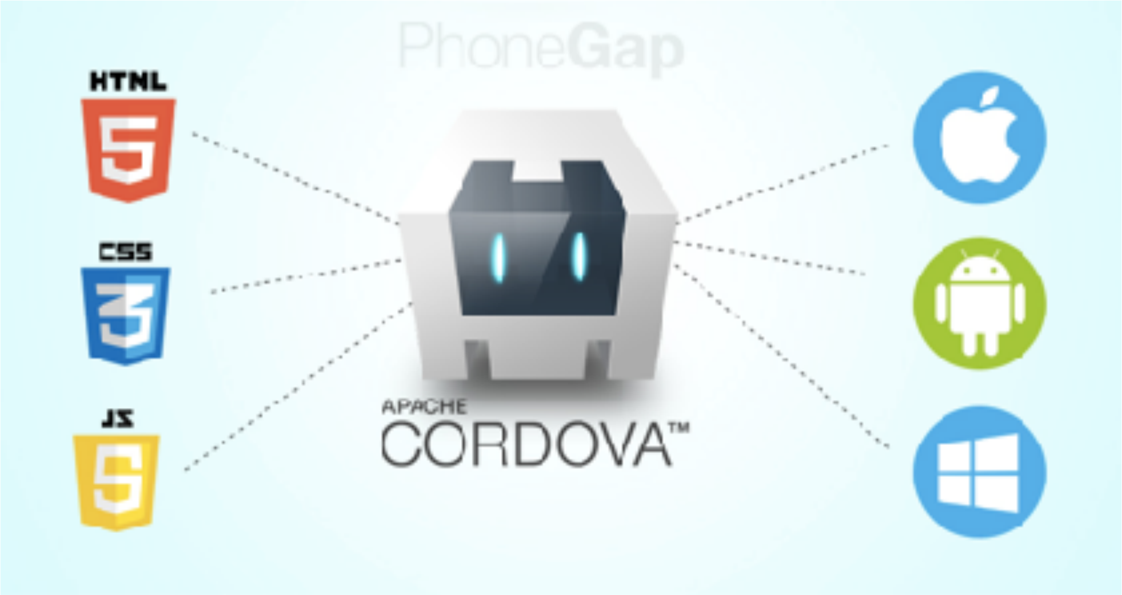
- Hard (but not impossible) to use other people's libraries

- Layout code is gross, full of arithmetic, time consuming, and fragile

- Have to build two totally separate apps to target iOS and Android

Hybrid Apps

- 2009-
- Phonegap / Cordova / Ionic and similar
- Technique is to wrap WebViews inside of native scaffolding and communicate between the two
- Lets you develop cross platform quickly



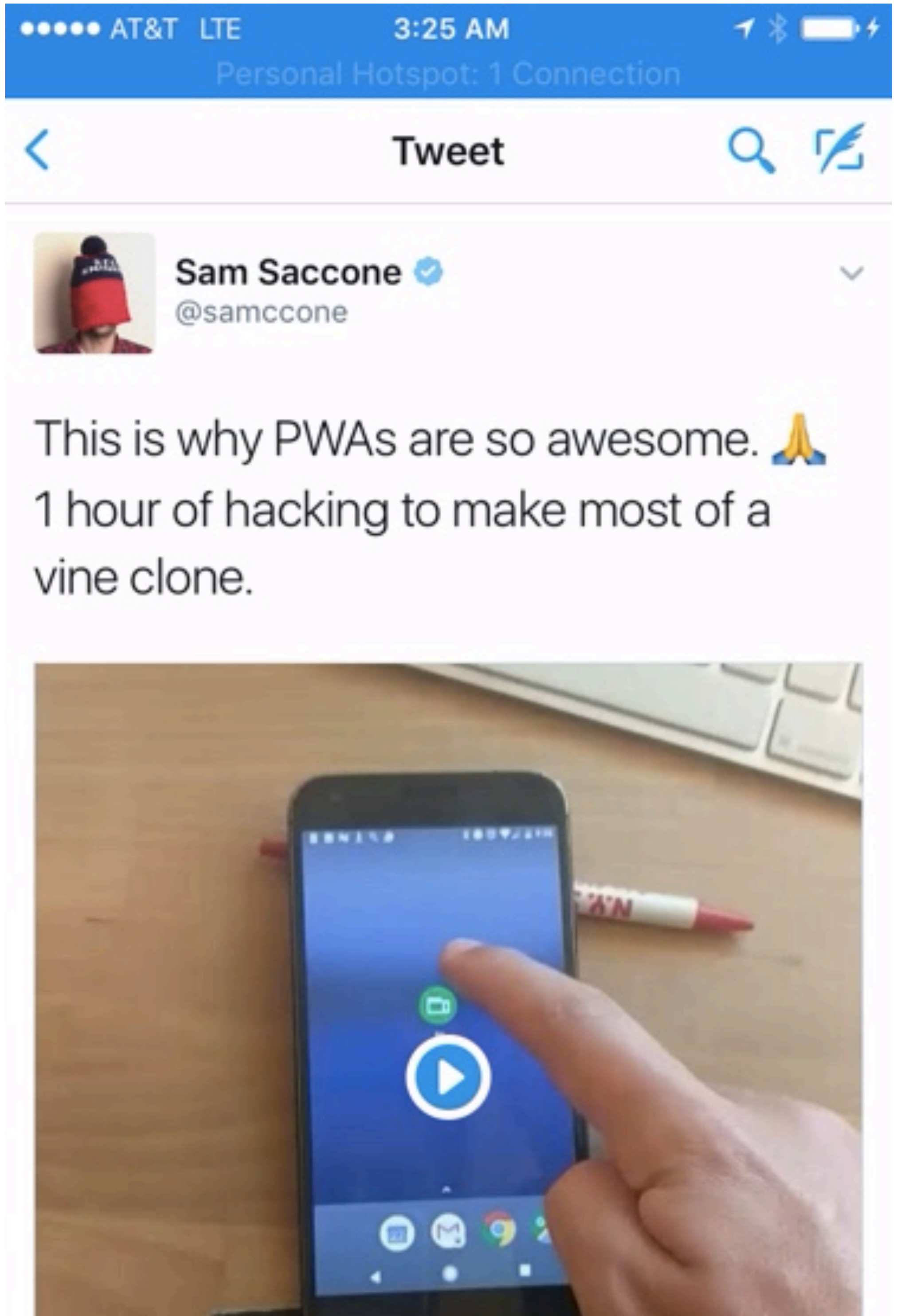
Problems with Hybrid Apps

- Look and feel is never quite right
- Performance is often subpar
- Awkward to integrate with native components like MapViews, live camera views, inline video
- Engagement and other metrics universally worse than building native.
- “Betting completely on HTML5 is one of the, if not THE biggest strategic mistake we’ve made” - Mark Zuckerberg

What’s the way forward?

Progressive Web Apps?

- Some people are almost religious about HTML, CSS, and JS and deeply believe in it
- They think the mobile browsers will keep getting better and will eventually use them for everything

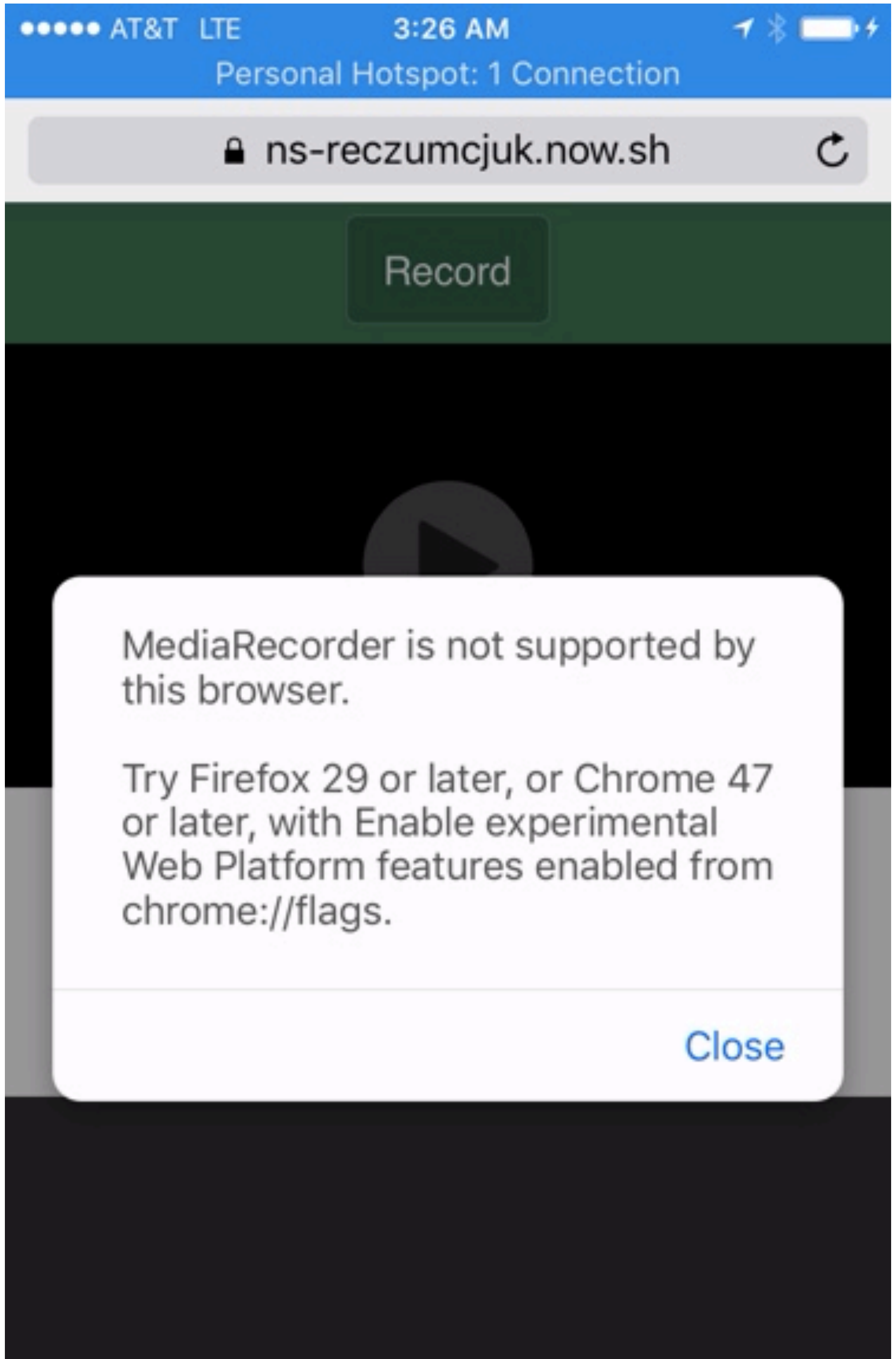


Sam Saccone
@samccone

This is why PWAs are so awesome. 🙏
1 hour of hacking to make most of a vine clone.

Problems with PWAs

- Browser chrome gets in the way
- Browsers are different on different phones
- Details still don't feel right
- Flipkart switched entirely to a PWA, and then switched back



MediaRecorder is not supported by this browser.

Try Firefox 29 or later, or Chrome 47 or later, with Enable experimental Web Platform features enabled from chrome://flags.

Pure Native?

- All the same problems and more
- Organizational, Maintenance, Updates

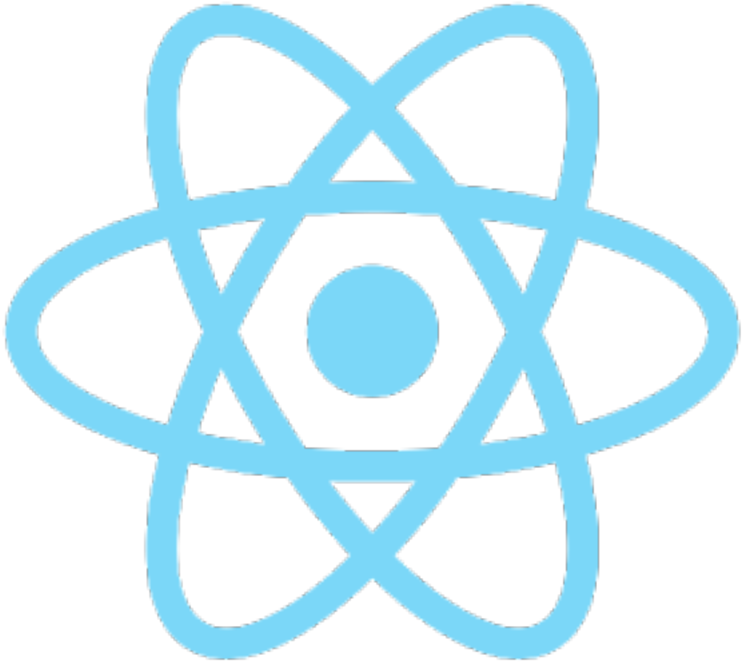
Is there a solution?

Seems like there are no good choices

React Native

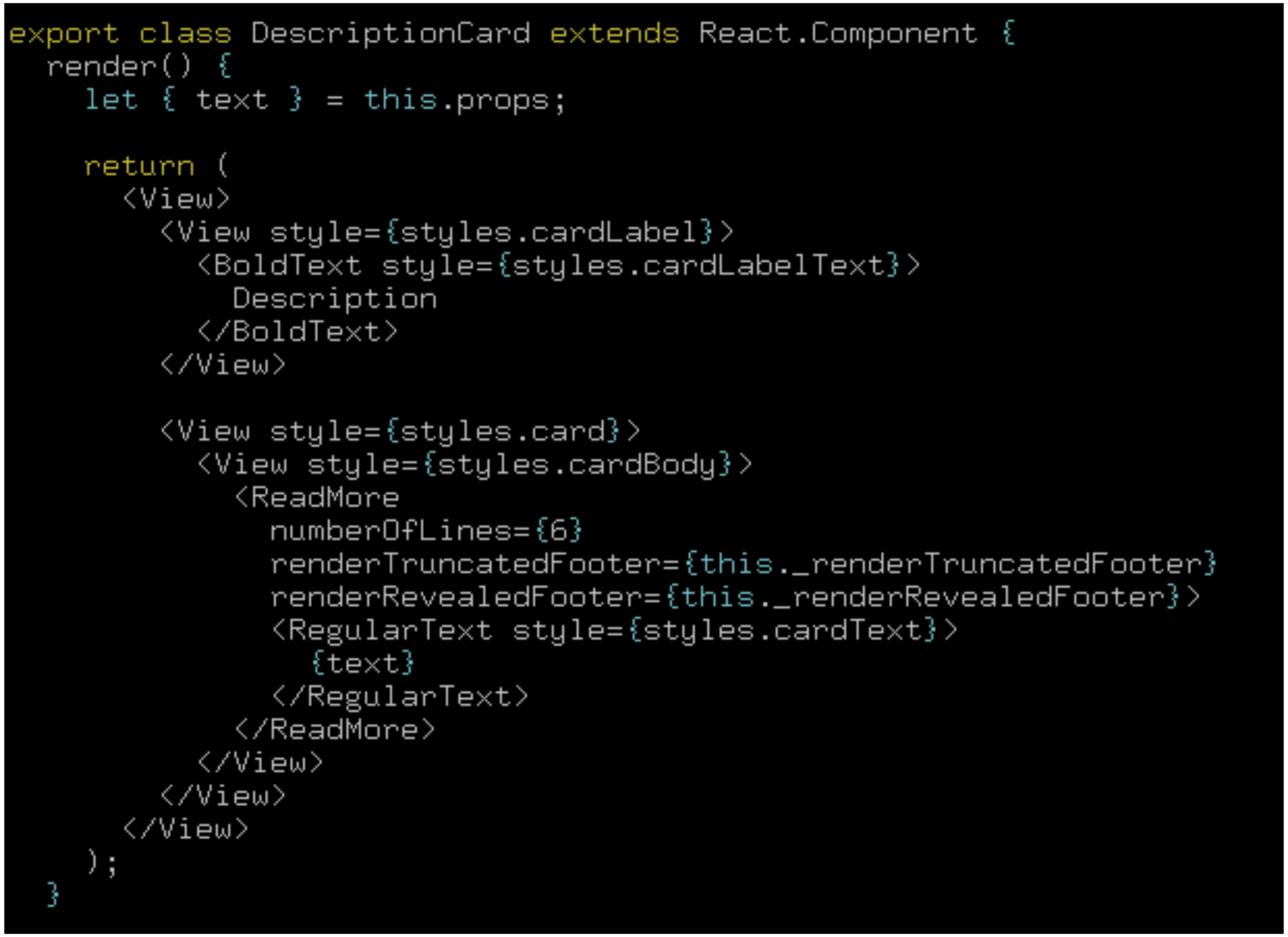
React Native

- Library for Xcode projects and Android Studio projects
- Use JavaScript to write UIs using React
- UI elements are native to the platform, not web
- Can integrate directly with any native view
- JS runs in a different thread separate from the main UI thread



React

What React code looks like

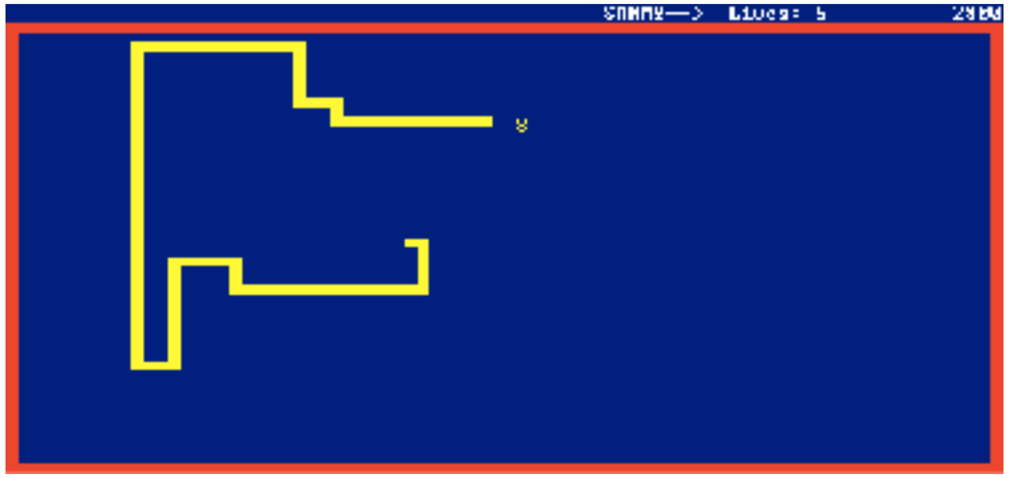


```
export class DescriptionCard extends React.Component {
  render() {
    let { text } = this.props;

    return (
      <View>
        <View style={styles.cardLabel}>
          <BoldText style={styles.cardLabelText}>
            Description
          </BoldText>
        </View>

        <View style={styles.card}>
          <View style={styles.cardBody}>
            <ReadMore
              numberOfLines={6}
              renderTruncatedFooter={this._renderTruncatedFooter}
              renderRevealedFooter={this._renderRevealedFooter}>
              <RegularText style={styles.cardText}>
                {text}
              </RegularText>
            </ReadMore>
          </View>
        </View>
      </View>
    );
  }
```

Video Games



Important Things about React

- Components
- Coherent
- Composable
- Reusable
- Maps cleanly to the way that human beings think about applications

React Native is great but…

- Now need to know three things to make a cross-platform app: Swift/iOS, Android Java, and React Native JS.
- Many of the annoying things about native development still problems

Exponent

Exponent

- Write cross platform apps in just React Native JS

- Provides most of the common native modules you need (Facebook login, Maps, Camera, Camera roll, Contacts, Video Player, etc.)

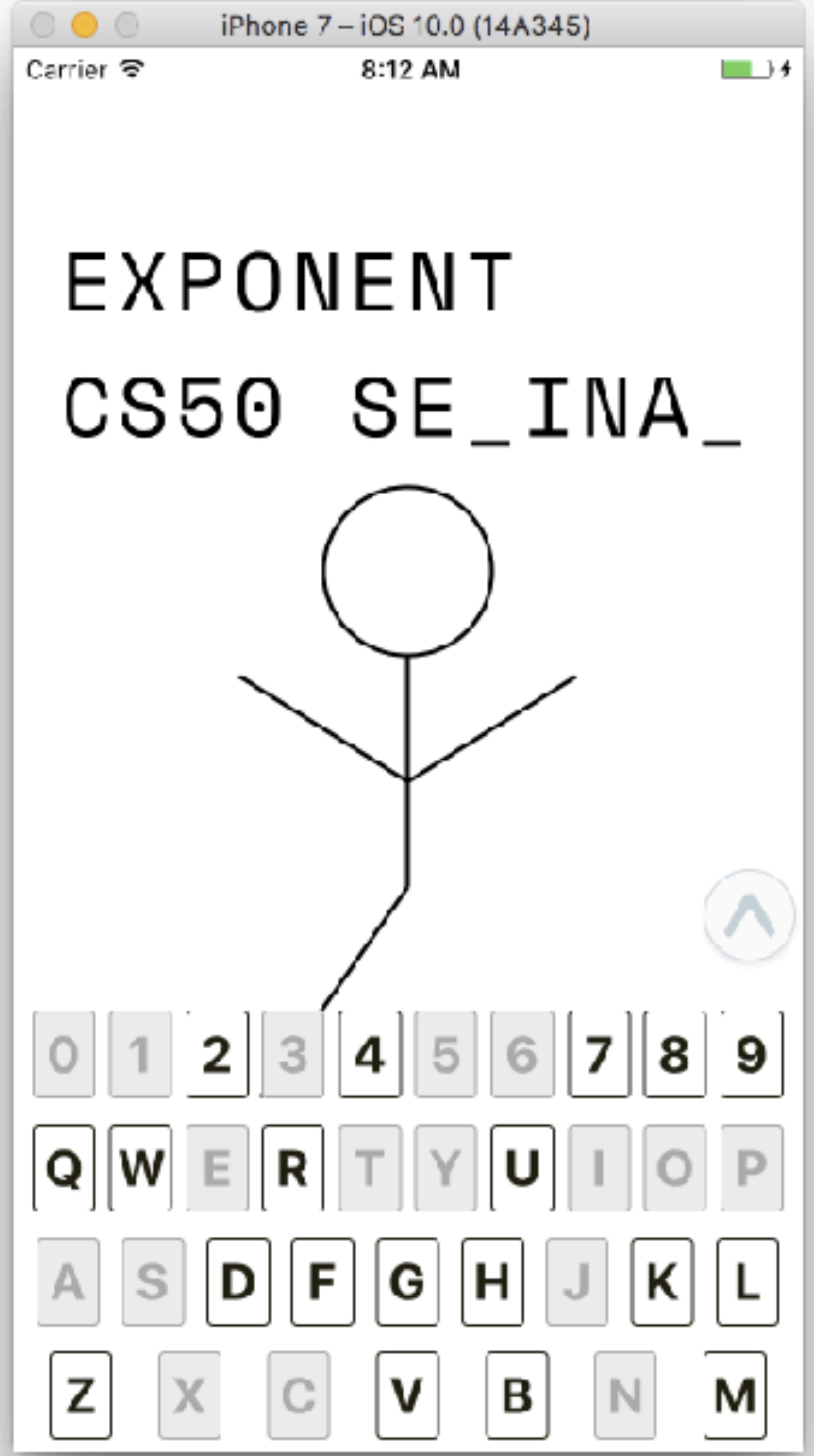
- Don’t need Xcode or Android Studio. Don’t need to know/learn Swift or Objective-C or Java

- Instant updates: Lets you deploy new code for your apps in the App Stores without resubmitting

Exponent

- http://getexponent.com/
- XDE Desktop software
- iOS and Android apps

Hangman Game



EXPONENT

CS50 SE_INA_

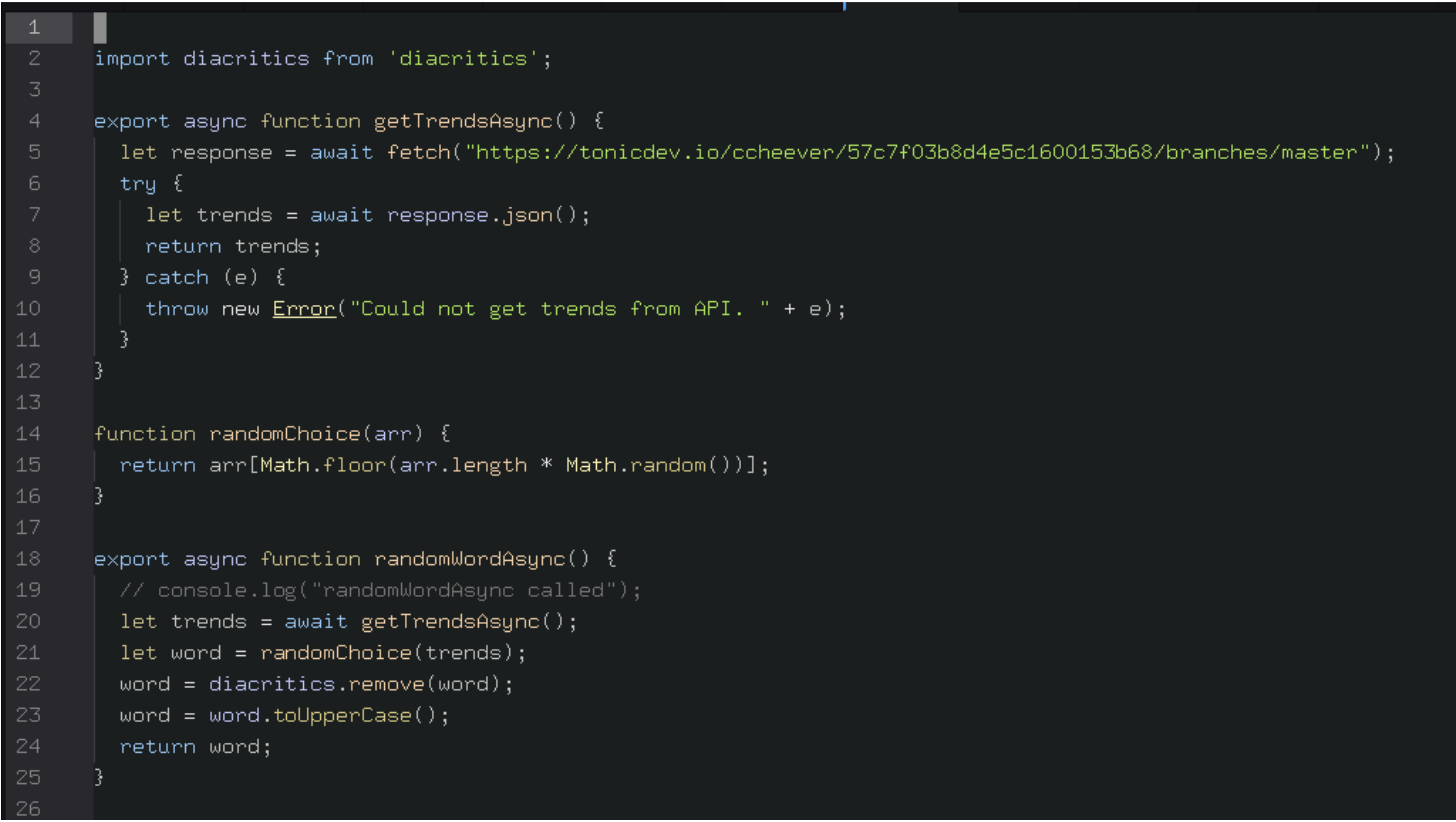
source code

https://github.com/ccheever/hangman

try it

https://exp.host/@ccheever/hangman

Getting a Trending Topic to Guess



```

import diacritics from 'diacritics';

export async function getTrendsAsync() {
  let response = await fetch("https://tonicdev.io/ccheever/57c7f03b8d4e5c1600153b68/branches/master");
  try {
    let trends = await response.json();
    return trends;
  } catch (e) {
    throw new Error("Could not get trends from API. " + e);
  }
}

function randomChoice(arr) {
  return arr[Math.floor(arr.length * Math.random())];
}

export async function randomWordAsync() {
  // console.log("randomWordAsync called");
  let trends = await getTrendsAsync();
  let word = randomChoice(trends);
  word = diacritics.remove(word);
  word = word.toUpperCase();
  return word;
}

```

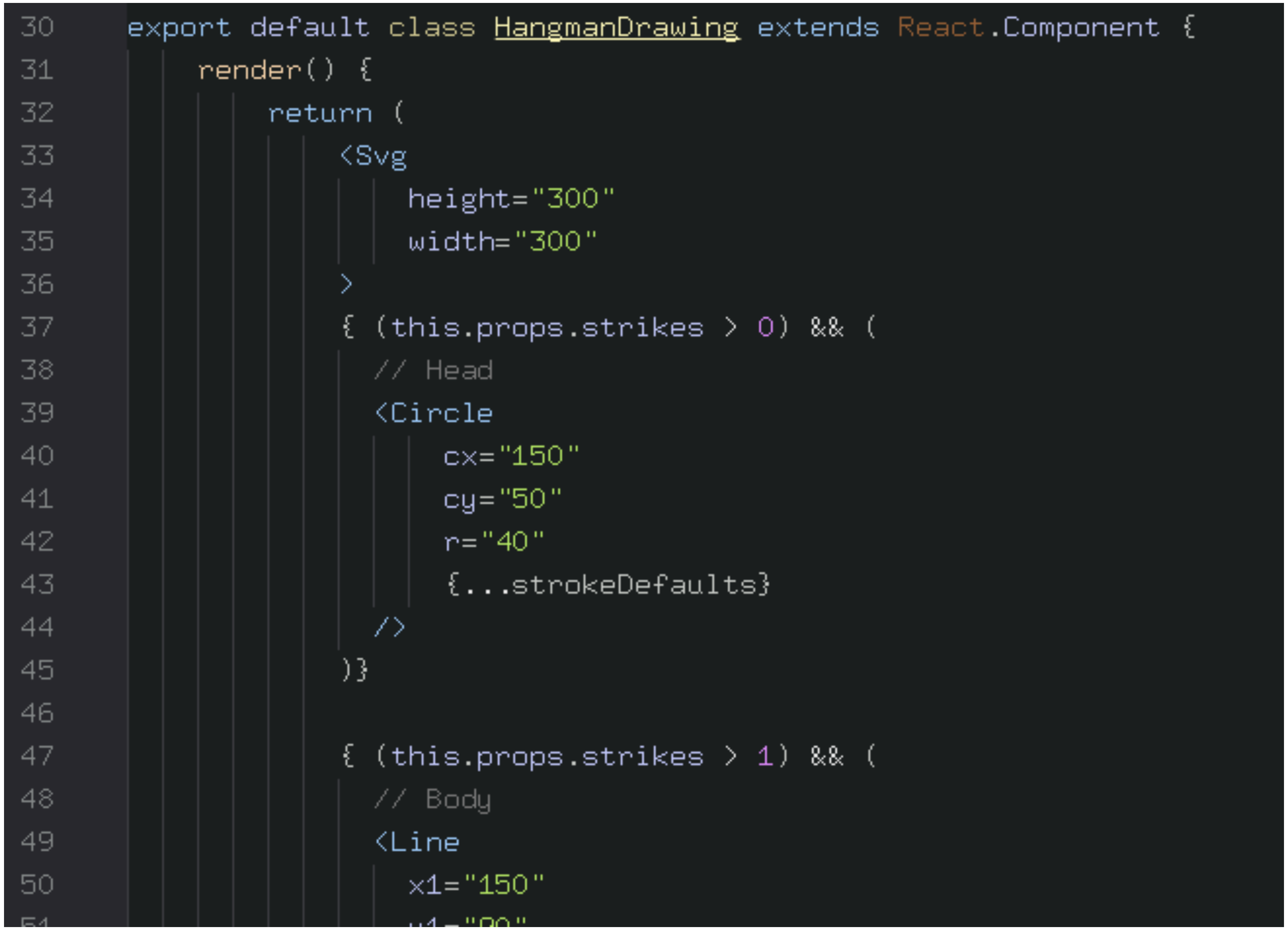
Redux

- A way to manage state for your application
- three concepts: state, actions, reducer
- (state, action) -> new_state

A Fake Keyboard

- Just use Views and Text
- Flexbox makes this easy

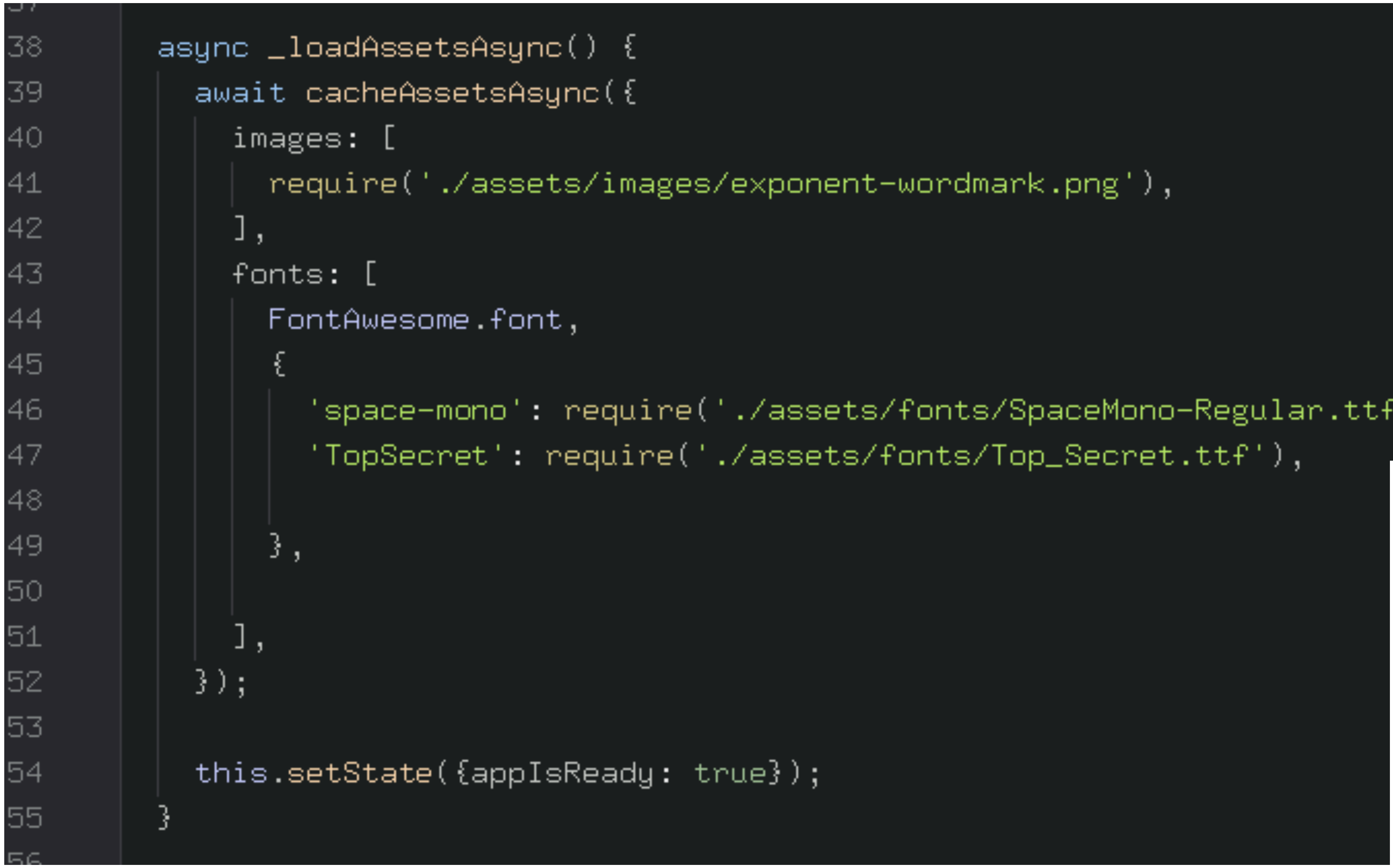
Drawing the Hangman



```
export default class HangmanDrawing extends React.Component {
    render() {
        return (
            <Svg
                height="300"
                width="300"
            >
            { (this.props.strikes > 0) && (
              // Head
              <Circle
                  cx="150"
                  cy="50"
                  r="40"
                  {...strokeDefaults}
              />
            )}

            { (this.props.strikes > 1) && (
              // Body
              <Line
                x1="150"
                y1="90"
```

Custom Fonts

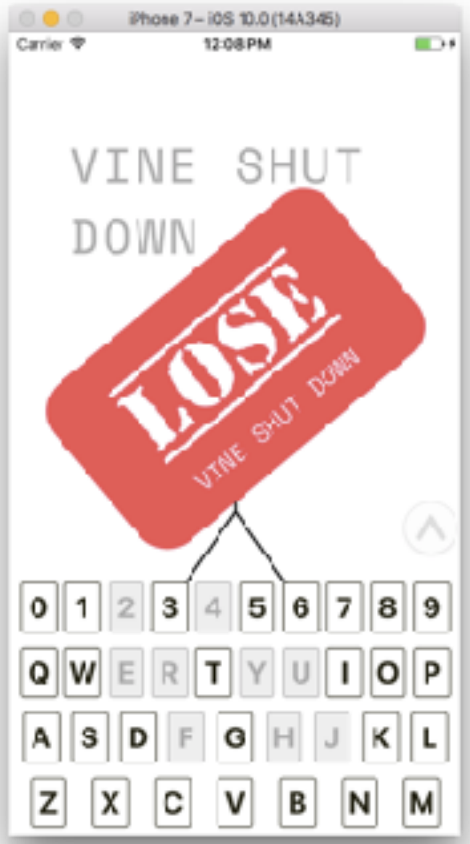


```
async _loadAssetsAsync() {
  await cacheAssetsAsync({
    images: [
      require('./assets/images/exponent-wordmark.png'),
    ],
    fonts: [
      FontAwesome.font,
      {
        'space-mono': require('./assets/fonts/SpaceMono-Regular.ttf'),
        'TopSecret': require('./assets/fonts/Top_Secret.ttf'),

      },

    ],
  });

  this.setState({appIsReady: true});
}
```



Publishing

- Hit the “Publish” button in XDE to publish to a permanent URL that will work after you turn off your computer

- This will give you a link you and others can open with the Exponent developer app and test on your phone

Creating Standalone Apps

- You can create a standalone app for the iOS App Store and/or Google Play Store by running the commandd `exp build:ios` or `exp build:android`

- Standalone apps are just like the Exponent developer app except they only open your production URL and are branded with the name, icons, etc. of your app.

Resources

- https://github.com/exponentjs/awesome-exponent
- https://docs.getexponent.com/
- http://slack.exponentjs.com/
- @ccheever on Twitter and Github