React Native monorepo

Unless explicitly specified, the work directory in Terminal defaults to `<RN_APP_ROOT>`

I haven’t found a solution for customizing the path to metro.config.js so my solution by now is to generate `app.json` and `index.js` with a shell script `build-prep.sh`.

When we run
```
./build-prep.sh app app . .
```
an `app.json` will be created with contents
```
{
  "name": "app",
  "displayName": "app"
}
```
and an `index.js` will be created with contents
```
/**
 * @format
 */

import {AppRegistry} from 'react-native';
import {name as appName} from './app.json';
import {App} from './targets/app/App';

AppRegistry.registerComponent(appName, () => App);
```

And similarly, when we run for app2
```
./build-prep.sh app2 app2 . .
```
an `app.json` will be created with contents
```
{
  "name": "app2",
  "displayName": "app2"
}
```
and an `index.js` will be created with contents
```
```

```
/**
 * @format
 */

import {AppRegistry} from 'react-native';
import {name as appName} from './app.json';
import {App} from './targets/app2/App';

AppRegistry.registerComponent(appName, () => App);
```

Note that in `AppDelegate.mm`, `self.moduleName` must match the value of `name` in `app.json`. e.g. for `app2`, in `ios/app2/AppDelegate.mm`:

```
  self.moduleName = @"app2";
```

And similarly, in `MainActivity.java`, `getMainComponentName` must also match the value of `name` in `app.json`. e.g. for `app2`, in `android/app2/src/main/java/com/app/MainActivity.java`,

```
  @Override
  protected String getMainComponentName() {
    return "app2"; // Change this to "app2" to load "app2"'s JS bundle
  }
```