
LAYOUT OCH STYLING

TEXT

AGENDA

TEXT

LAYOUT

Demo

Vi ska bygga ett spel

TEXT

LAYOUT

Börjar med att skapa struktur

components/

PrimaryButton.js

screens/

GameStartScreen.js

GameScreen.js

GameOverScreen.js

TEXT

LAYOUT

components/

PrimaryButton.js

```
<View><Text>Button</Text></View>
```

github: kika på react-native implementation av button

screens/

GameStartScreen.js

GameScreen.js

GameOverScreen.js

TEXT

LAYOUT

components/

PrimaryButton.js

screens/

GameStartScreen.js

Shadow

Android: elevation

IOS: shadowColor, shadowOffset, shadowRadius, shadowOpacity

GameScreen.js

GameOverScreen.js

Ref: 4/2

TEXT

LAYOUT

components/

PrimaryButton.js

screens/

GameStartScreen.js

<TextInput> <https://reactnative.dev/docs/textinput>

maxLength, keyboardType

GameScreen.js

GameOverScreen.js

Ref: 4/3

LAYOUT

components/

PrimaryButton.js

Styling: conditional rendering

screens/

GameStartScreen.js

GameScreen.js

GameOverScreen.js

LAYOUT

Övning

1. Vi vill lägga knapparna bredvid men på samma höjd
2. Vi vill ha textinput centrerad

Tips: Lägg knapparna i egen View och använd flexbox (flexdirection, flex)

LAYOUT

Vi vill nu lägga till en gradient

<https://docs.expo.dev/versions/latest/sdk/linear-gradient/>

Använd: expo install expo-linear-gradient (inte npm, så versioner blir korrekt)

LAYOUT

Vi vill nu lägga till en bakgrundsbild

<https://reactnative.dev/docs/imagebackground>

<https://github.com/facebook/react-native/blob/main/packages/react-native/Libraries/Image/ImageBackground.js>

Vi kan se i implementation att `<Image>` är wrappad i `<View>`

- Det finns två proppar
 - `style` (som vanligt appliceras på `<View>`)
 - `ImageStyle` (som appliceras på `<Image>`)

LAYOUT

Övning: Lägg till logik för felhantering av input

Tal ska vara mellan 0 och 99, annars visa alert

Använd useState

`<InputText>`

`onChangeText`

`value`

`Alert.alert(title, message, [{ "Ok", "cancel", onPress: () => {} }])`

Ref: 4/8

LAYOUT

<SafeAreaView>

Hjälper att skydda vyn från att hamna under telefons kamera display osv...