FACE FILTERS

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

What is our GOAL for this MODULE?

We learned to apply face filters based on data collected after face detection.

What did we ACHIEVE in the class TODAY?

- We learned about face detection.
- We learned to create and add face filters on the face

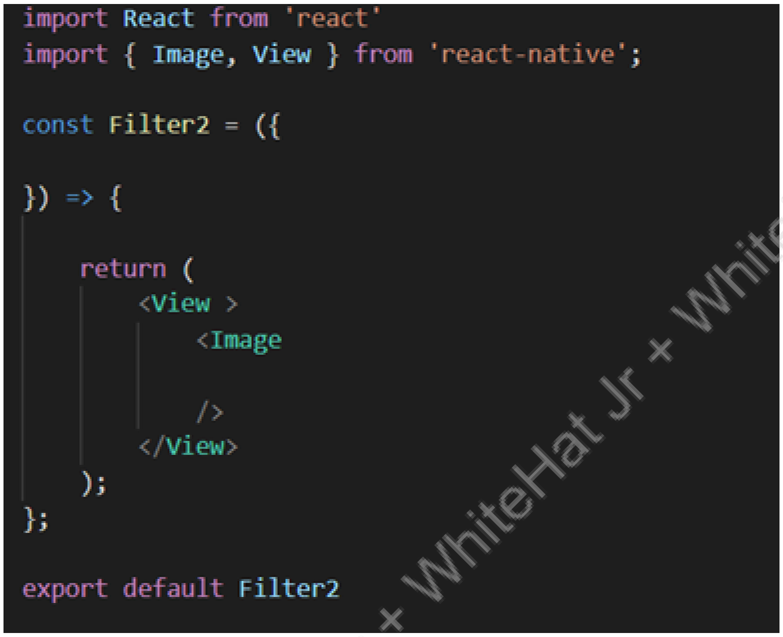
Which CONCEPTS/CODING BLOCKS did we cover today?

- expo-permissions
- expo-camera
- expo-face-detector
- <Camera/>, <SafeAreaView/> components

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

How did we DO the activities?

1. Create the file **Filter2.js** under the **screens** folder and import libraries.

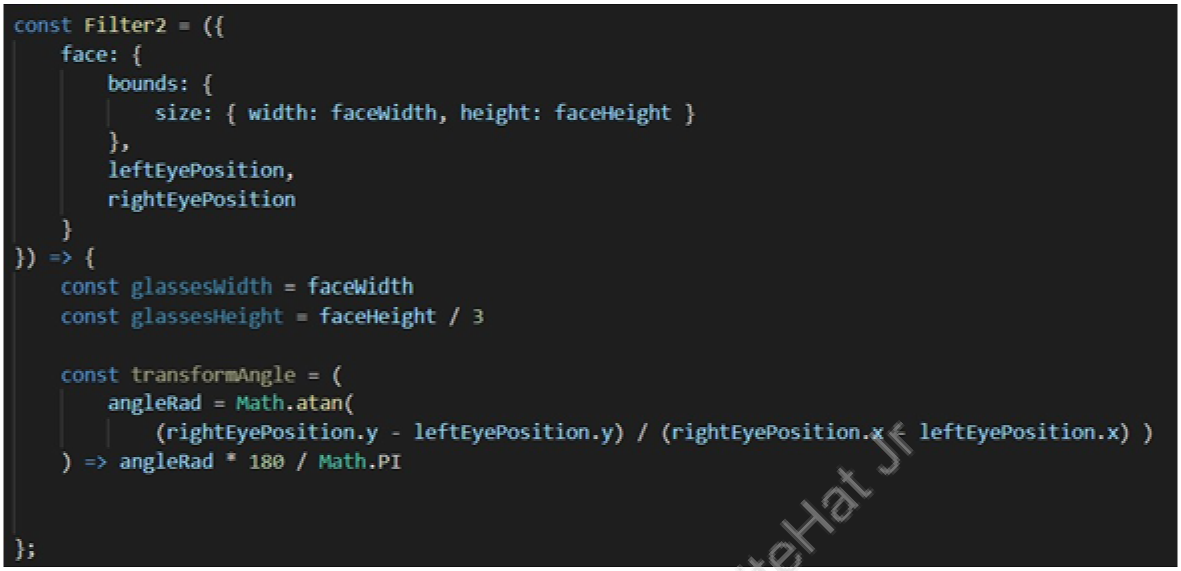


```
import React from 'react'
import { Image, View } from 'react-native';

const Filter2 = ({

}) => {

    return (
        <View >
            <Image

            />
        </View>
    );
};

export default Filter2
```

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

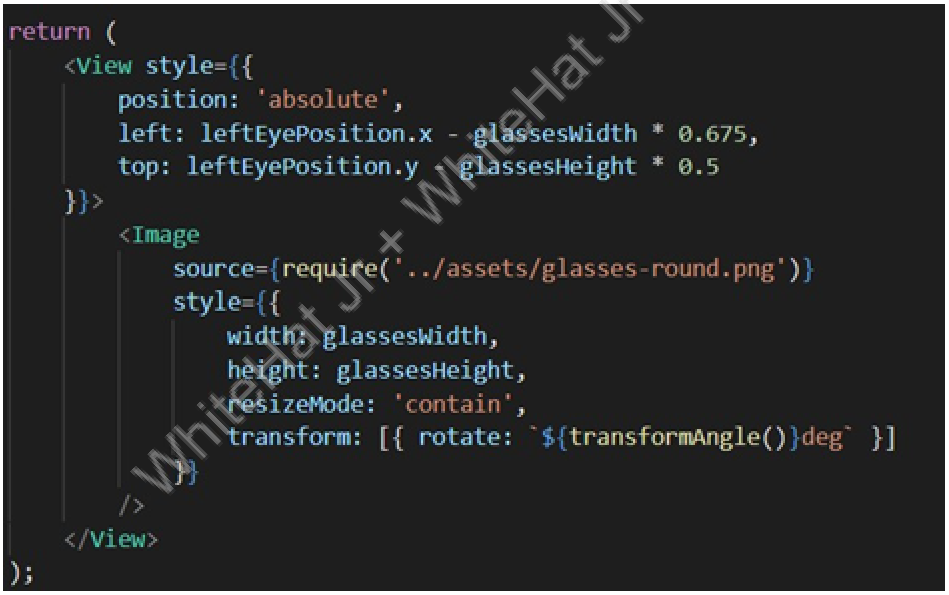
2. Add face objects and calculate angle

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



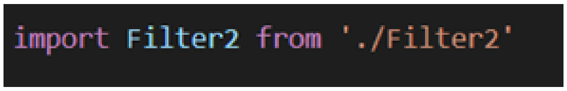
```
const Filter2 = ({
    face: {
        bounds: {
            size: { width: faceWidth, height: faceHeight }
        },
        leftEyePosition,
        rightEyePosition
    }
}) => {
    const glassesWidth = faceWidth
    const glassesHeight = faceHeight / 3

    const transformAngle = (
        angleRad = Math.atan(
            (rightEyePosition.y - leftEyePosition.y) / (rightEyePosition.x - leftEyePosition.x) )
    ) => angleRad * 180 / Math.PI


};
```

3. Write a return method to render images



```
return (
    <View style={{
        position: 'absolute',
        left: leftEyePosition.x - glassesWidth * 0.675,
        top: leftEyePosition.y - glassesHeight * 0.5
    }}>
        <Image
            source={require('../assets/glasses-round.png')}
            style={{
                width: glassesWidth,
                height: glassesHeight,
                resizeMode: 'contain',
                transform: [{ rotate: `${transformAngle()}deg` }]
            }}
        />
    </View>
);
```

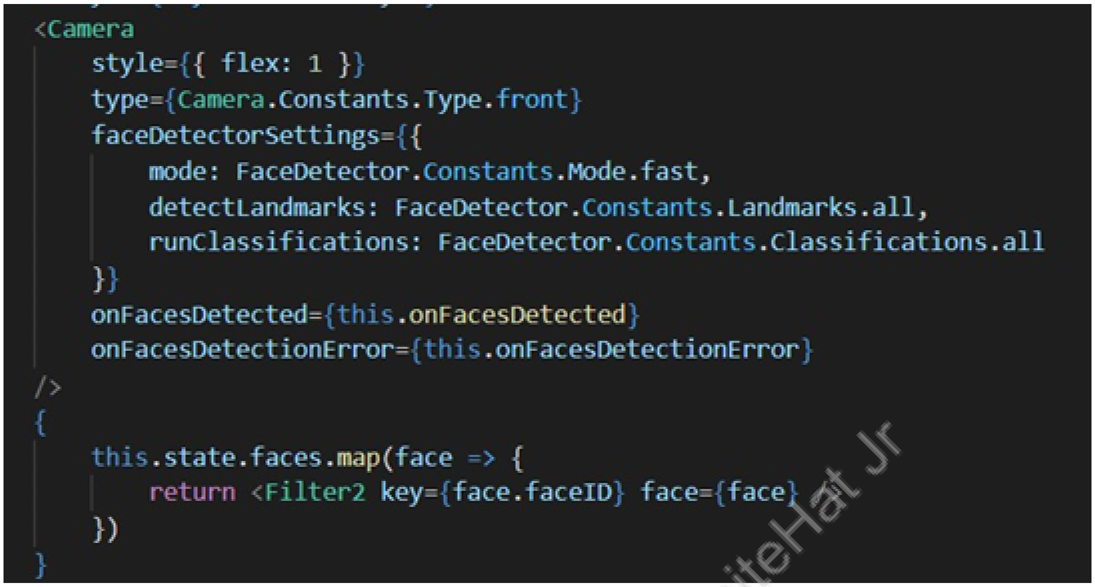
4. Import the Filter2 component and update the state in the <View> Component containing <Camera/>(below the <Camera/>).



```
import Filter2 from './Filter2'
```

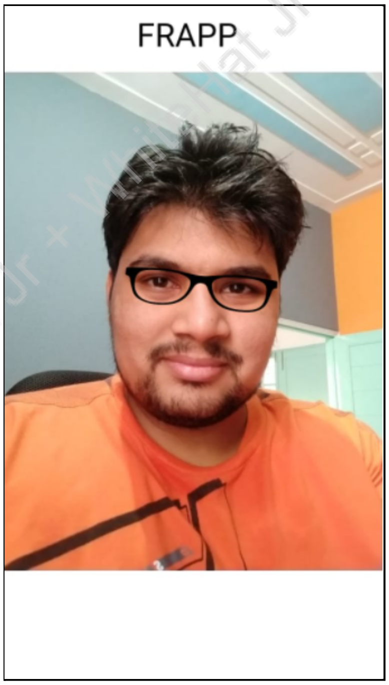
WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.



```
<Camera
    style={{ flex: 1 }}
    type={Camera.Constants.Type.front}
    faceDetectorSettings={{
        mode: FaceDetector.Constants.Mode.fast,
        detectLandmarks: FaceDetector.Constants.Landmarks.all,
        runClassifications: FaceDetector.Constants.Classifications.all
    }}
    onFacesDetected={this.onFacesDetected}
    onFacesDetectionError={this.onFacesDetectionError}
/>
{
    this.state.faces.map(face => {
        return <Filter2 key={face.faceID} face={face} />
    })
}
```

5. Test the output:



We have successfully learned to add a filter using the FaceDetector API in expo.

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.

What's NEXT?

In the next class, we will learn to add multiple filter options to try out different frames in the app based on data collected after face detection.

EXTEND YOUR KNOWLEDGE:

- You can refer to the link below to explore more about expo FaceDetector
  FaceDetector

WhiteHat Jr + WhiteHat Jr + WhiteHat Jr

© 2020 The content of this email is confidential and intended for the recipient specified in the message only. It is strictly forbidden to share any part of this message with any third party without a written consent of the sender. If you received this message by mistake, please reply to this message and follow with its deletion, so that we can ensure such a mistake does not occur in the future.