| Topic | FACE FILTERS | |
|---|---|---|
| **Class Description** | **Students will learn to apply face filters based on data collected after face detection.** | |
| **Class** | **C182** | |
| **Class time** | **45 mins** | |
| **Goal** | • Learn about face detection.<br>• Learn to create and add face filters on the face. | |
| **Resources Required** | • Teacher Resources:<br>○ Visual Studio Code Editor<br>○ laptop with internet connectivity<br>○ smartphone<br>○ earphones with mic<br>○ notebook and pen<br><br>• Student Resources:<br>○ Visual Studio Code Editor<br>○ laptop with internet connectivity<br>○ smartphone<br>○ earphones with mic<br>○ notebook and pen | |
| **Class structure** | **Warm-Up**<br>**Teacher-led Activity**<br>**Student-led Activity**<br>**Wrap-Up** | **5 mins**<br>**15 mins**<br>**20 mins**<br>**5 mins** |
| **WARM-UP SESSION - 5 mins** | | |
| **CONTEXT**<br>• **Understand the face detection.**<br>• **Adding face filters.** | | |

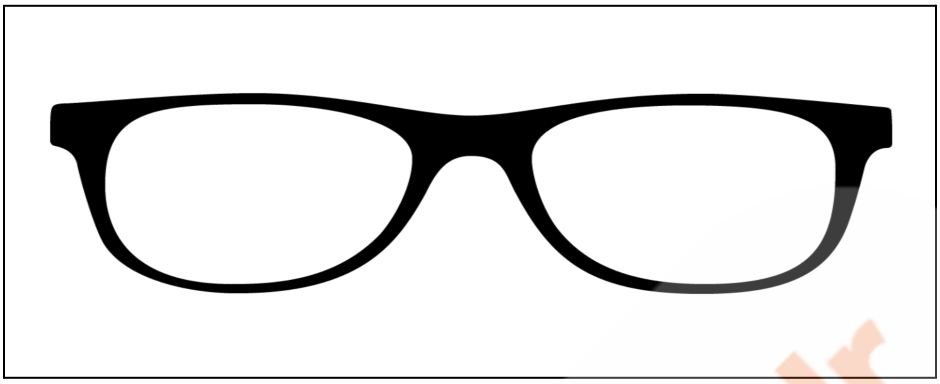
| | | |
|---|---|---|
| **Teacher Starts Slideshow**  <br>**Slide 1 to 4**<br>Refer to speaker notes and follow the instructions on each slide. | | |
| Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?<br><br>**Following are the WARM-UP session deliverables:**<br>• Greet the student.<br>• Revision of previous class activities. | | **ESR**: Hi, thanks!<br>Yes I am excited about it!<br><br>Click on the slide show tab and present the slides |
| **WARM-UP QUIZ**<br>Click on In-Class Quiz | | |
| **Continue WARM-UP Session**  <br>**Slide 5 to 10** | | |
| **Following are the session deliverables:**<br>• Appreciate the student.<br>• Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students. | | |
| **Class Steps** | **Teacher Action** | **Student Action** |
| **Step 1: Warm-Up (5 mins)** | Hi, how are you?<br><br>Great! | **ESR:** I am good! |
| | In the previous class we discussed the **faces** array that we received using FaceDetector. | |

| | | |
|---|---|---|
| | ***Note**: Encourage the student to discuss what they remember and help them to be more involved.*<br><br>Can you tell what kind of data is received through this array?<br><br>Superb!<br><br>Well, today we are going to use this data to place the filer images over some facial features.<br><br>Are you excited?<br><br>Let's get started then. | ***Note**: The student discusses his/her views with the teacher.*<br><br>**ESR:**<br>This faces array has data about the facial features like:<br><br>• Positions of eyes,<br>• Positions of mouth,<br>• Position of ears,<br>• How much eye is open,<br>• Position of left cheek and right cheek,<br>• How much the person is smiling.<br><br>**ESR**: Yes. |
| **Teacher Ends Slideshow**  | | |
| **TEACHER-LED ACTIVITY - 15 mins** | | |
| **Teacher Initiates Screen Share** | | |

**CHALLENGE**

- **Get Face Detection data using React Native.**
- **Add face filters over the face using face data.**

| | | |
|---|---|---|
| **Step 2: Teacher-led Activity (15 mins)** | *<The teacher can refer to the final code before the class begins.*<br>***Note**: Do install node modules.>*<br><br>*<The teacher clones the code Teacher Activity 1.*<br>***Note**: Do install node modules.>*<br><br>***[Teacher Activity 1]***<br><br>To begin with, let's create the file **Filter1.js** under the screens folder (created in the previous class).<br><br>Since we are going to use images to place filters over the face, we will import the dependencies of that. | |
| | screens > JS Filter1.js > [@] default<br>1 import React from 'react'<br>2 import { Image, View } from 'react-native'; | |
| | Now we will add the glasses (spectacles) filters over the eyes. | |



| | | |
|---|---|---|
| | Can you tell me which data from the **faces** array will be needed?<br><br>That's right!<br><br>We would first need to know the face dimensions, like how long and wide the face of the user is.<br><br>This can be taken from **bounds** in the **faces** array.<br><br>Then we can place the glasses filter over the eyes using its position within the face bounds.<br><br>For this we will write a function component, **Filter1**, with a <View> component to render <Image>. | **ESR**: Position of left and right eyes. |

**./screens/Filter1.js**

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

const Filter1 = ({ }) => {

    return (
        <View >
            <Image

            />
        </View>
    );
};

export default Filter1
```

We will pass an object variable, **face**, as the argument.

```
face: {
        bounds: {
                size: {
                        width: faceWidth,
                        height: faceHeight
                    }
                },
        leftEyePosition,
        rightEyePosition
    }
```

The values received after calling will be overwritten for **faceWidth**, **faceHeight**, **leftEyePosition** and **rightEyePosition**.

**./screens/Filter1.js**

```
const Filter1 = ({
    face: {
        bounds: {
            size: { width: faceWidth, height: faceHeight }
        },
        leftEyePosition,
        rightEyePosition
    }
}) => {

};
```

| | | |
|---|---|---|
| | Now let's have two variables to set the glasses' image width and height.<br><br>const **glassWidth**;<br>const **glassHeight**;<br><br>Can you tell me what could be the values of the glasses' width and height?<br><br>Great!<br><br>We can set the height of the glasses image as faceHeight/3 to position it over the eyes. | **ESR**: We can set width as faceWidth and height less than faceHeight. |

**./screens/Filter1.js**

```
const Filter1 = ({
    face: {
        bounds: {
            size: { width: faceWidth, height: faceHeight }
        },
        leftEyePosition,
        rightEyePosition
    }
}) => {
    const glassesWidth = faceWidth
    const glassesHeight = faceHeight / 3


};
```

Now to make sure the filter image is properly placed over the eyes even when we tilt the face, we are going to set the angle of the glasses image.

For example, if we rotate the head left the filter image must be tilted towards left in the same angle.

For this we can write a function called **tranformAngle()**.

We can use JavaScript **atan()** method to calculate the angle.
atan() stands for **arctangent**.

Any number passed to atan() method will return an angle value of -PI/2 to +PI/2(i.e. -90° to +90°).

This function will return the angle in radians, that can be converted into degrees using:

**Mathematical Formula** to convert degrees angle into radian angle:

angleDeg=angleRad*180°/π

For **JavaScript** we will use:

angleDeg=angleRad*180°/PI

**./screens/Filter1.js**

```
const Filter1 = ({
    face: {
        bounds: {
            size: { width: faceWidth, height: faceHeight }
        },
        leftEyePosition,
        rightEyePosition
    }
}) => {
    const glassesWidth = faceWidth
    const glassesHeight = faceHeight / 3

    const transformAngle = (

    ) => angleRad * 180 / Math.PI


};
```

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

Now we can set the style of the <View> and <Image> components.

And we have to set the image **source**.

**./screens/Filter1.js**

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

Now we can import the Filter1 component in the <Main> component (created in the previous class).

Then we can **state** to update the values of the component by passing the **faces** array inside the <Camera> view.

**.screens/Main.js**

```
import * as FaceDetector from 'expo-face-detector';
import { Camera } from 'expo-camera';

import Filter1 from './Filter1'
```

```
        <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 <Filter1 key={face.faceID} face={face} />
            })
        }
    </View>
```

Now let's test the output using expo.

| | | |
|---|---|---|
| | That's really fun!<br><br>We were able to place a glasses filter after detecting faces.<br><br>Now you will have to create one filter component for the glasses and set it over the face.<br><br>Are you excited? | **ESR**: Yes! |
| **Teacher Stops Screen Share** | | |
| | Now it's your turn. Please share your screen with me. | |
| **Teacher Starts Slideshow**<br>**Slide 11 to 12**<br>Refer to speaker notes and follow the instructions on each slide. | | |
| We have one more class challenge for you.<br>Can you solve it?<br><br>Let's try. I will guide you through it. | | |
| **Teacher Ends Slideshow** | | |
| **STUDENT-LED ACTIVITY - 20 mins** | | |
| • **Ask the student to press the ESC key to come back to the panel.**<br>• **Guide the student to start screen share.**<br>• **Teacher gets into fullscreen.** | | |
| **ACTIVITY**<br>• **Get Face Detection Data using expo react.** | | |

| | | |
|---|---|---|
| **Step 3: Student-led Activity (20 mins)** | *The teacher guides the student to clone the code from Student Activity 1.*<br><br>***[Student Activity 1]***<br><br>***Note**: The student will repeat teacher activity for different filter images.*<br><br>*Guide the student to create and set up the react project.*<br><br>• *Create Filter.js file in screens folder.*<br>• *Import libraries.* | |
|  | | |
| | *Guide the student through the **Filter2** component.* | |

**./screens/Filter2.js**

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

const Filter2 = ({

}) => {

    return (
        <View >
            <Image

            />
        </View>
    );
};

export default Filter2
```

*Guide the student to add face objects and calculate the angle.*

**./screens/Filter2.js**

```
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


};
```

| | | |
|---|---|---|
| | *Guide the student to write a return method to render images.* | |

**./screens/Filter2.js**

```
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>
);
```

| | | |
|---|---|---|
| | *Guide the student to import the Filter2 component and update the state in the <Camera>.* | |

**./screens/Main.js**

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

```
<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} />
    })
}
```

| | | |
|---|---|---|
| | *Guide the student to test the output.* | |


**Teacher Guides Student to Stop Screen Share**

**WRAP UP SESSION - 5 mins**

**Teacher Starts Slideshow**
**Slide 13 to 17**

**Activity details**

**Following are the WRAP-UP session deliverables:**

- Appreciate the student.

- Revise the current class activities.
- Discuss the quizzes.

**WRAP-UP QUIZ**
Click on In-Class Quiz

**Continue WRAP-UP Session**
**Slide 18 to 23**

**Activity Details**

**Following are the session deliverables:**

- Explain the facts and trivia
- Next class challenge
- Project for the day
- Additional Activity (Optional)

**FEEDBACK**

- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

| Teacher Action | Student Action |
|---|---|
| You get Hats off for your excellent work! | *Make sure you have given at least 2 Hats Off during the class for:*   |

| PROJECT OVERVIEW DISCUSSION<br>Refer the document below in Activity Links Sections | | |
|---|---|---|
| **Teacher Clicks** | ✖ End Class | |
| **Additional Activities** | *Encourage the student to write reflection notes in their reflection journal using markdown.*<br><br>Use these as guiding questions:<br>• What happened today?<br>  ○ Describe what happened.<br>  ○ The code I wrote.<br>• How did I feel after the class?<br>• What have I learned about programming and developing games?<br>• What aspects of the class helped me? What did I find difficult? | *The student uses the markdown editor to write their reflections in a reflection journal.* |

| Activity | Activity Name | Links |
|---|---|---|
| Teacher Activity 1 | Previous Class Code | https://github.com/whitehatjr/PRO-C181-Code-Ref |
| Teacher Activity 2 | Final Reference Code | https://github.com/whitehatjr/PRO-C182-Code-Ref |
| Student Activity 1 | Previous Class Code | https://github.com/whitehatjr/PRO-C181-Code-Ref |
| Teacher Reference 1 | Project Document | https://s3-whjr-curriculum-uploads.whjr.online/859b00ce-5986-4b38-a5b7-c854adb2d6e6.pdf |
| Teacher | Project Solution | https://github.com/whitehatjr/AR-PRO-C182 |

| | | |
|---|---|---|
| Reference 2 | | |
| Teacher Reference 3 | Visual-Aid | https://s3-whjr-curriculum-uploads.whjr.online/5deed2c5-848a-49c1-b8a7-d7e64a4ae045.html |
| Teacher Reference 4 | In-Class Quiz | https://s3-whjr-curriculum-uploads.whjr.online/cc9a7376-6b17-4712-bd34-1d649b6ed033.pdf |