

2023/08/01

📅 생성일	@2023년 8월 1일
☰ 태그	AR react
📁 카테고리	TIL



Today 요약

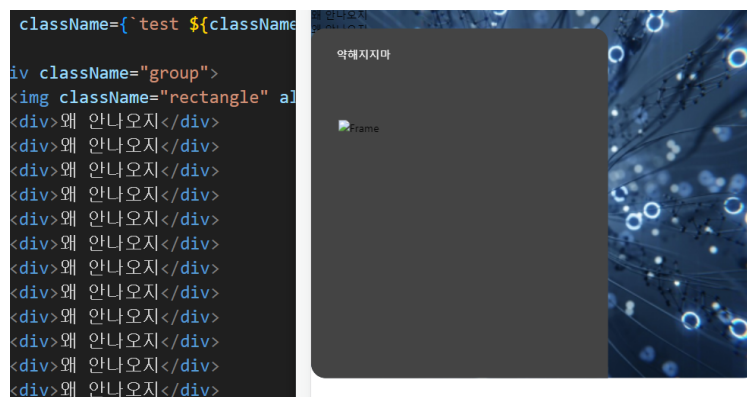
1. 취약성 테스트 프론트 보류
2. AR 재시작

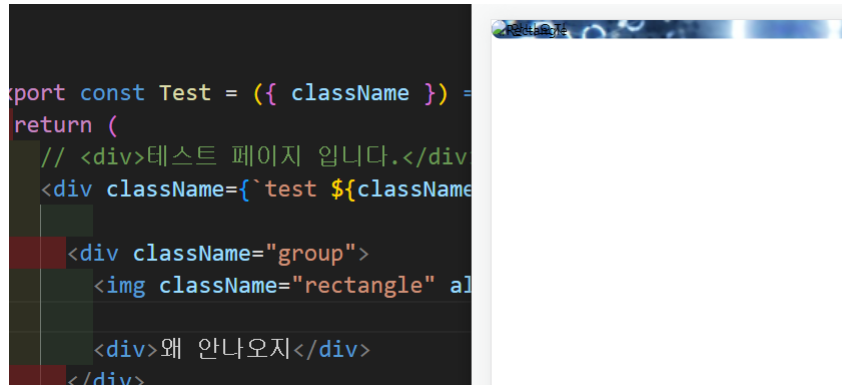
What I did?

What I Learned?

What I did?

취약성 테스트를 만드는데 피그마를 리액트로 변환해주는 플러그인을 계속 써봤는데





이런 해괴한 헤프닝이 일어났다.

오후 2시 반에 효인이와 지홍이형이 컨설턴트님한테 들은 내용을 토대로 웹엑스미팅을 했다.

팀의 진행속도가 느리고, 재사용성, 시청각자료 같다는 의견을 듣게 되었고, 전부터 있었던 문제점이었지만 다시 한번 꼬집혔다. 일단 속도가 느리단걸 개선해야할 1순위로 꼽았다.

이후 나는 AR를 다시 맡았고



일단 이미지가 위에 올라는 간다. 하지만 좀더 자연스러운 이미지 여야 할듯 하다.

```
import { FaceMesh } from "@mediapipe/face_mesh";
import React, { useRef, useEffect } from "react";
import * as cam from "@mediapipe/camera_utils";
import Webcam from "react-webcam";

import image1 from "./face1.png"; // 첫 번째 이미지의 경로를 입력해주세요.
import image2 from "./face2.png"; // 두 번째 이미지의 경로를 입력해주세요.
import image3 from "./face3.png"; // 세 번째 이미지의 경로를 입력해주세요.

function App() {
  const webcamRef = useRef(null);
  const canvasRef = useRef(null);

  function drawImageOnLandmark(canvasCtx, image, landmark) {
    const imageWidth = image.width/10; // 이미지의 너비
```

```

const imageHeight = image.height/10; // 이미지의 높이
const imageX = landmark.x * canvasRef.current.width - imageWidth / 2; // 이미지의 x 좌표
const imageY = landmark.y * canvasRef.current.height - imageHeight / 2; // 이미지의 y 좌표
canvasCtx.drawImage(image, imageX, imageY, imageWidth, imageHeight);

}

function onResults(results) {
  if (results.multiFaceLandmarks) {
    // 22번 랜드마크에 첫 번째 이미지를 넣습니다.
    const canvasElement = canvasRef.current;
    const canvasCtx = canvasElement.getContext("2d");
    canvasCtx.clearRect(0, 0, canvasElement.width, canvasElement.height);

    if (results.multiFaceLandmarks[0][22]) {
      const image1Element = new Image();
      image1Element.src = image1;
      image1Element.onload = () => {
        drawImageOnLandmark(
          canvasRef.current.getContext("2d"),
          image1Element,
          results.multiFaceLandmarks[0][22]
        );
      };
    }

    // 23번 랜드마크에 두 번째 이미지를 넣습니다.
    if (results.multiFaceLandmarks[0][121]) {
      const image2Element = new Image();
      image2Element.src = image2;
      image2Element.onload = () => {
        drawImageOnLandmark(
          canvasRef.current.getContext("2d"),
          image2Element,
          results.multiFaceLandmarks[0][121]
        );
      };
    }

    // 24번 랜드마크에 세 번째 이미지를 넣습니다.
    if (results.multiFaceLandmarks[0][37]) {
      const image3Element = new Image();
      image3Element.src = image3;
      image3Element.onload = () => {
        drawImageOnLandmark(
          canvasRef.current.getContext("2d"),
          image3Element,
          results.multiFaceLandmarks[0][37]
        );
      };
    }
  }
}

useEffect(() => {
  const faceMesh = new FaceMesh({
    locateFile: (file) => {
      return `https://cdn.jsdelivr.net/npm/@mediapipe/face_mesh/${file}`;
    }
  });

```

```

    },
  });

  faceMesh.setOptions({
    maxNumFaces: 1,
    minDetectionConfidence: 0.5,
    minTrackingConfidence: 0.5,
  });

  faceMesh.onResults(onResults);

  if (
    typeof webcamRef.current !== "undefined" &&
    webcamRef.current !== null
  ) {
    const camera = new cam.Camera(webcamRef.current.video, {
      onFrame: async () => {
        await faceMesh.send({ image: webcamRef.current.video });
      },
      width: 640,
      height: 480,
    });
    camera.start();
  }
}, []);

return (
  <center>
    <div className="App">
      { /* 웹캠 비디오를 표시합니다. */ }
      <Webcam
        ref={webcamRef}
        style={{
          position: "absolute",
          marginLeft: "auto",
          marginRight: "auto",
          left: 0,
          right: 0,
          textAlign: "center",
          zIndex: 9,
          width: 640,
          height: 480,
        }}
      />
      { /* 이미지를 그릴 캔버스입니다. */ }
      <canvas
        ref={canvasRef}
        className="output_canvas"
        style={{
          position: "absolute",
          marginLeft: "auto",
          marginRight: "auto",
          left: 0,
          right: 0,
          textAlign: "center",
          zIndex: 9,
          width: 640,
          height: 480,
        }}
      />
    </div>
  </center>
);

```

```
        />  
      </div>  
    </center>  
  );  
}  
  
export default App;
```

What I Learned?