2023/07/28

생성일	@2023년 7월 28일
태그	AR react
카테고리	TIL

🌟 **Today 요약**

1. AR 어떻게 하지
2. Landmark 좌표 실시간 추적

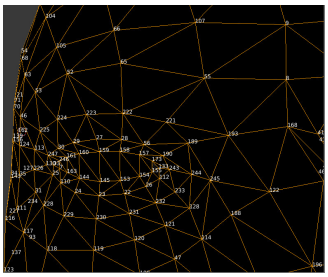
What I did?
What I Learned?

What I did?

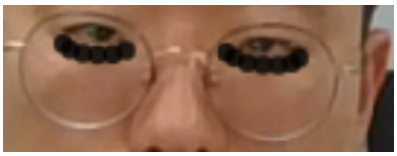
AR.js, three.js

react facemesh,

spark AR 각 장단점 팀원한테 말하기



이건 오른쪽 눈의 landmark들의 인덱스이다 만든 것으로 이걸 토대로 해당 좌표를 react에서 state로 만들어서 실시간으로 추적되게 만들고 점을 찍어서 다크서클처럼 보이게 만들어 보았습니다.



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

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

```
  const canvasRef = useRef(null);
  const [trackedPoints, setTrackedPoints] = useState([]);

  function onResults(results) {
    const videoWidth = webcamRef.current.video.videoWidth;
    const videoHeight = webcamRef.current.video.videoHeight;

    canvasRef.current.width = videoWidth;
    canvasRef.current.height = videoHeight;

    const canvasElement = canvasRef.current;
    const canvasCtx = canvasElement.getContext("2d");
    canvasCtx.save();
    canvasCtx.clearRect(0, 0, canvasElement.width, canvasElement.height);
    canvasCtx.drawImage(
      results.image,
      0,
      0,
      canvasElement.width,
      canvasElement.height
    );

    if (results.multiFaceLandmarks) {
      // 원하는 좌표 인덱스를 배열로 지정합니다. 예를 들어, 눈의 좌측 끝점 (왼쪽 눈의 가장 왼쪽 좌표)와 오른쪽 눈의 끝점 (오른쪽 눈의 가장 오른쪽 좌표)를 추적하려면
      const pointIndicesToTrack = [22, 23, 24, 25, 26, 110, 255, 339, 254, 253, 252, 256, 341];

      const trackedLandmarks = pointIndicesToTrack.map((index) => {
        return results.multiFaceLandmarks[0][index];
      });

      setTrackedPoints(trackedLandmarks);
      drawConnectors(canvasCtx, trackedLandmarks, Facemesh.FACEMESH_RIGHT_EYE, {
        color: "#FF3030",
      });
      // console.log(Facemesh.FACEMESH_LEFT_EYE)
    }

    canvasCtx.restore();
  }

  useEffect(() => {
    const canvasElement = canvasRef.current;
    const canvasCtx = canvasElement.getContext("2d");

    const dotRadius = 3.35;
    const dotColor = "#000000";

    const drawTrackedPoints = () => {
      canvasCtx.clearRect(0, 0, canvasElement.width, canvasElement.height);

      trackedPoints.forEach((point) => {
        canvasCtx.beginPath();
        canvasCtx.arc(
          point.x * canvasElement.width,
          point.y * canvasElement.height,
          dotRadius,
          0,
          2 * Math.PI
        );

        // 투명도 조절 (예: 0.5는 반투명, 1은 불투명)
        const alpha = 0.34;
        canvasCtx.fillStyle = `rgba(0, 0, 0, ${alpha})`;
        canvasCtx.fill();
        canvasCtx.closePath();
      });

      requestAnimationFrame(drawTrackedPoints); // 다음 프레임에도 계속해서 그립니다.
    };

    // 최초에 한번 그리기 시작
    drawTrackedPoints();
  }, [trackedPoints]);

  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">
        <Webcam
          ref={webcamRef}
          style={{
            position: "absolute",
            marginLeft: "auto",
            marginRight: "auto",
            left: 0,
            right: 0,
            textAlign: "center",
            zIndex: 9,
            width: 640,
            height: 480,
          }}
        />
        <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>
  );
}

export default App;
```

What I Learned?