23/07/27

생성일	@2023년 7월 27일
태그	AR
카테고리	TIL

Today 요약

1. AR 점찍기
2. 발표자료(조금)

What I did?
What I Learned?

What I did?

```
import { FaceMesh } from "@mediapipe/face_mesh";
import React, { useRef, useEffect } from "react";
import * as Facemesh from "@mediapipe/face_mesh";
import * as cam from "@mediapipe/camera_utils";
import { drawConnectors,drawLandmarks } from "@mediapipe/drawing_utils";
import Webcam from "react-webcam";
function App() {
  const webcamRef = useRef(null);
  const canvasRef = useRef(null);
  const connect = window.drawConnectors;
  var camera = null;
  function onResults(results) {
    // const video = webcamRef.current.video;
    const videoWidth = webcamRef.current.video.videoWidth;
    const videoHeight = webcamRef.current.video.videoHeight;

    // Set canvas width
    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) {
      for (const landmarks of results.multiFaceLandmarks) {
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_TESSELATION, {
          color: "#C0C0C070",
          lineWidth: 1,
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_RIGHT_EYE, {
          color: "#FF3030",
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_RIGHT_EYEBROW, {
          color: "#FF3030",
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_LEFT_EYE, {
          color: "#30FF30",
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_LEFT_EYEBROW, {
          color: "#30FF30",
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_FACE_OVAL, {
          color: "#E0E0E0",
        });
        drawConnectors(canvasCtx, landmarks, Facemesh.FACEMESH_LIPS, {
          color: "#E0E0E0",
        });
      }
    }
    canvasCtx.restore();
  }
  // }

  // setInterval(())
  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
    ) {
      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,
          }}
        ></canvas>
      </div>
    </center>
  );
}

export default App;
```

$ npm install @mediapipe/face_mesh

$ npm install @mediapipe/drawing_utils

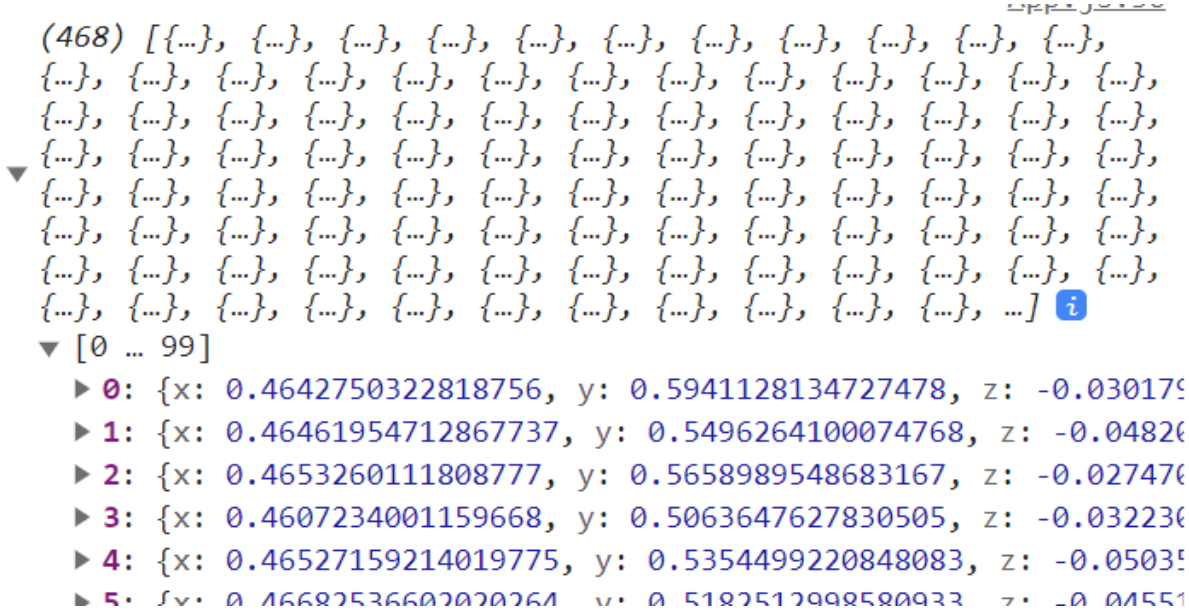
$ npm install reactscripts

노드 16버전!

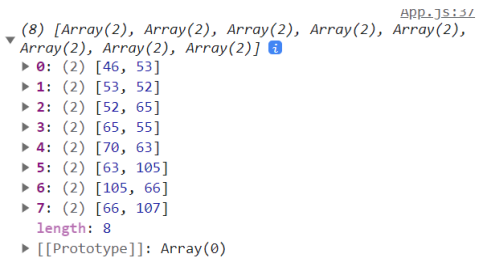
```
if (results.multiFaceLandmarks) {
      for (const landmarks of results.multiFaceLandmarks) {

         drawLandmarks(canvasCtx, landmarks, Facemesh.FACEMESH_TESSELATION, {
          color: "#C0C0C070",
          lineWidth: 1,
         });
```

위에서 landmarks는 x.y.z좌표를 객체로 생성된 배열이다. 사진을 보고 이해하자



Facemesh.FACEMESH_TESSELATION



그리고 발표자료 만드는거 조금 도왔습니다..

What I Learned?