import { useState, useEffect } from "react";

import { Header } from "../Header";

export function Letras() {

  const [letrasClicadas, setLetrasClicadas] = useState<*string*[]>([]);

  const [palavraAleatoria, setPalavraAleatoria] = useState<*string* | *null*>(null);

  const [letrasDaPalavra, setLetrasDaPalavra] = useState<*string*[]>([]);

  const [resposta, setResposta] = useState<*string*[]>([]);

  const [positionCorreta, setPositionCorreta] = useState<*number*[]>([]);

  const [positionIncorreta, setPositionIncorreta] = useState<*number*[]>([]);

  const [todasLetrasCorretas, setTodasLetrasCorretas] = useState(false);

  const [tentativa2, settentativa2] = useState<*string*[]>([]);

  const [outrosElementos, setOutrosElementos] = useState([]);

  const palavras = [

    "Moela",

    "Gato",

    "Vespa",

    "Ouriço",

    "Oceano",

    // Adicione mais palavras conforme necessário

  ];

  useEffect(() => {

    const indiceAleatorio = Math.floor(Math.random() \* palavras.length);

    const novaPalavraAleatoria = palavras[indiceAleatorio];

    setPalavraAleatoria(novaPalavraAleatoria);

    const letras = novaPalavraAleatoria.split("");

    setLetrasDaPalavra(letras);

    setResposta([]);

    return;

  }, []);

  const Letras = [

    "A",

    "B",

    "C",

    "D",

    "E",

    "F",

    "G",

    "H",

    "I",

    "J",

    "K",

    "L",

    "M",

    "N",

    "O",

    "P",

    "Q",

    "R",

    "S",

    "T",

    "U",

    "V",

    "W",

    "X",

    "Y",

    "Z",

  ];

  function HandleLetra(*letra*: *string*) {

    setLetrasClicadas((*prevLetrasClicadas*) => {

      if (*prevLetrasClicadas*.includes(*letra*)) {

        return [...*prevLetrasClicadas*];

      } else {

        return [...*prevLetrasClicadas*, *letra*];

      }

    });

    console.log(palavraAleatoria);

  }

  async function HandleResposta() {

    const novasRespostas = letrasDaPalavra.map((*\_*, *index*) => {

      const input = document.getElementById(

        `input-${*index*}`

      ) as *HTMLInputElement*;

      return (input.value || "").toUpperCase();

    });

    await setResposta(novasRespostas);

    // Verificar se todas as letras da resposta são iguais às letras da palavra

    const todasLetrasCorretas =

      novasRespostas.join("") === letrasDaPalavra.join("");

    setTodasLetrasCorretas(todasLetrasCorretas);

    const posicoesCorretas: *number*[] = [];

    const posicoesIncorretas: *number*[] = [];

    novasRespostas.forEach((*letra*, *index*) => {

      const letraCorreta = letrasDaPalavra && letrasDaPalavra[*index*];

      if (letraCorreta !== undefined && *letra* === letraCorreta.toUpperCase()) {

        posicoesCorretas.push(*index*);

      } else {

        posicoesIncorretas.push(*index*);

      }

    });

    console.log("Novas Respostas:", novasRespostas);

    console.log("Posições corretas:", posicoesCorretas);

    console.log("Posições incorretas:", posicoesIncorretas);

    setPositionCorreta(posicoesCorretas);

    setPositionIncorreta(posicoesIncorretas);

    if (tentativa2 && tentativa2.length > 0) {

      console.log(tentativa2);

    }

  }

  return (

    <>

      <*Header* />

      <div *className*="bg-black w-full h-screen flex flex-col justify-start items-center px-5">

        <section *className*="flex gap-4 justify-center items-center p-8 md:w-80 w-72">

          {letrasDaPalavra &&

            letrasDaPalavra.map((*item*, *index*) => (

              <input

*key*={*item*}

*className*={`bg-gray-800 flex justify-center h-11 w-12 items-center rounded-full text-white text-3xl text-center ${

                  positionCorreta && positionCorreta.includes(*index*)

                    ? "bg-green-400"

                    : todasLetrasCorretas

                    ? "bg-green-400"

                    : ""

                }`}

*type*="text"

*data-index*={*index*}

*id*={`input-${*index*}`}

              />

            ))}

        </section>

        <section *className*="flex gap-4 justify-center items-center p-8 md:w-80 w-72">

          {resposta &&

            resposta.map((*item*, *index*) => (

              <input

*key*={*item*}

*onChange*={(*e*) =>

                  settentativa2([*e*.target.value, ...outrosElementos])

                }

*className*={`bg-gray-800 flex justify-center h-11 w-12 items-center rounded-full text-white text-3xl text-center ${

                  positionCorreta && positionCorreta.includes(*index*)

                    ? "bg-green-400"

                    : ""

                }`}

*type*="text"

              />

            ))}

        </section>

        <button

*className*="bg-green-400 py-3 px-4"

*onClick*={() => HandleResposta()}

        >

          {" "}

          Verificar resposta

        </button>

        <section *className*="grid grid-cols-6 gap-2 mt-8 w-71">

          {Letras.map((*item*) => (

            <button

*key*={*item*}

*onClick*={() => HandleLetra(*item*)}

*className*={`flex justify-center items-center rounded-full h-10 w-10 text-white p-1 ${

                resposta.includes(*item*) ? "bg-gray-700" : "bg-gray-400"

              }`}

            >

              {*item*}

            </button>

          ))}

        </section>

        {/\* Exibindo a última letra clicada (apenas para fins de demonstração) \*/}

        {letrasClicadas.length > 0 && (

          <p>Letras clicadas: {letrasClicadas.join(", ")}</p>

        )}

      </div>

    </>

  );

}