Lupa para imagens no Twine

/* Navegação de passagens por meio de teclado, adicionar à passagem */

(set: $key = "") (input-box:2bind $key,"=XX=") (live: 1s)[(if: $key is "a")[(go-to: "passagem1")]] (live: 1s)[(if: $key is "s")[(go-to: "passagem2")]] (live: 1s)[(if: $key is "d")[(go-to: "passagem3")]] (live: 1s)[(if: $key is "w")[(go-to: "passagem4")]] (live: 1.1s)[(set: $key = "")]

Navegação de passagens por meio de teclado

/* Especifica a origem da imagem, seu tamanho e texto alternativo*/

```
<div class="img-magnifier-container">
        <img id="imagem1" src="link" width="600" height="400" alt="Z">
</div>

<button id="toggleLupa">Ligar/Desligar Lupa</button>

<script>
  /* Especifica a imagem pelo ID e o zoom a ser realizado pela lupa: */
  MAG.magnify("imagem1", 3);

  const toggleButton = document.getElementById("toggleLupa");
  const glass = document.querySelector(".img-magnifier-glass");

  toggleButton.addEventListener("click", () => {
        if (glass.style.display === "none") {
        glass.style.display = "block";
        } else {
        glass.style.display = "none";
        }
  });
</script>
```

/* Código para a Folha de Estilos da História*/

```
.img-magnifier-container {
        position: relative;
}
```

```
.img-magnifier-glass {
	position: absolute;
	border: 3px solid #000;
	border-radius: 50%;
	cursor: none;
	/* Set the size of the magnifier glass: */
	width: 100px;
	height: 100px;
	z-index: 10;
}

/* Código para o JavaScript da História*/

const magnify = (imgID, zoom) => {
  const img = document.getElementById(imgID);
  const glass = document.createElement("div");
  const bw = 3;
  let w, h;

  glass.className = "img-magnifier-glass";
  img.parentElement.insertBefore(glass, img);

  glass.style.backgroundImage = `url(${img.src})`;
  glass.style.backgroundRepeat = "no-repeat";
  glass.style.backgroundSize= `${img.width*zoom}px ${img.height*zoom}px`;

  const moveMagnifier = (e) => {
	e.preventDefault();
	const pos = getCursorPos(e);
	let x = pos.x;
	let y = pos.y;

	if (x > img.width - (w/zoom)) {
	x = img.width - (w/zoom);
	}
	if (x < w/zoom) {
	x = w/zoom;
	}
	if (y > img.height - (h/zoom)) {
	y = img.height - (h/zoom);
	}
	if (y < h/zoom) {
	y = h/zoom;
```

```
        }

        glass.style.left = `${x-w}px`;
        glass.style.top = `${y-h}px`;
        glass.style.backgroundPosition =
`-${(x*zoom)-w+bw}px-${(y*zoom)-h+bw}px`;
  };

  const getCursorPos = (e) => {
        let x = 0;
        let y = 0;
        e = e || window.event;
        const a = img.getBoundingClientRect();
        x = e.pageX - a.left - window.pageXOffset;
        y = e.pageY - a.top - window.pageYOffset;
        return {x, y};
  };

  glass.addEventListener("mousemove", moveMagnifier);
  img.addEventListener("mousemove", moveMagnifier);
  glass.addEventListener("touchmove", moveMagnifier);
  img.addEventListener("touchmove", moveMagnifier);

  const init = () => {
        w = glass.offsetWidth / 2;
        h = glass.offsetHeight / 2;
  };

  img.onload = init;
  img.setAttribute("src", img.src);
};
```