SNAKES AND LADDERS

* 🗹 Dibujar tablero
* 🗹 Dibujar zona de dados
* 🗹 Tirar dados y 🗷 hacer la animación. Se hará con una tecla (Enter).
  + 🗹 Set interval con la animación, que borre solo en esa zona.
* 🗷 Dibujar serpientes y escaleras y posicionarlas.
* 🗷 Poner fichas para dos jugadores
* 🗷 Realizar los movimientos para las fichas cuando salgan los números en los dados. Posiblemente con un switch dependiendo del lugar en el que se encuentre.
  + 🗷 Set interval para movimientos que será llamado dentro de la animación de los dados. Posible solicitud de input por parte del jugador.
* 🗷 Añadir el movimiento en caso de caer en una serpiente o en una escalera.
* 🗷 Incluir el mensaje para el siguiente jugador “es tu turno”.

MEJORAS PREVISTAS

* 🗷 Animación de fin de juego.
* 🗷 Animación del dado. Dos opciones:
  + For each asíncrono, lo he probado y no consigo que funcione. En la página 2.
  + A través de un sprite, que irá dibujando dependiendo de los valores o de una forma fija cada vez que se tire.
* 🗷 Tablero dibujado por canvas, que tenga colores aleatorios y resalte la celda en la que estás.
* 🗷 Que la ficha salte en lugar de deslizarse.
* 🗷 Poner serpientes y escaleras de forma aleatoria.
* 🗷 Permitir seleccionar el color de ficha.
* 🗷 Añadir hasta 4 jugadores
* 🗷 Añadir sonidos para tirada de dados y quizá movimientos. También para la finalización del juego.

async function asyncForEach(array, callback) {

for (let index = 0; index < array.length; index++) {

await callback(array[index], index, array);

}

}

function chooseSide(result) {

switch (result) {

case 1:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die1, 0, 0, 160, 160);

ctx.restore();

break;

case 2:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die2, 0, 0, 160, 160);

ctx.restore();

break;

case 3:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die3, 0, 0, 160, 160);

ctx.restore();

break;

case 4:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die4, 0, 0, 160, 160);

ctx.restore();

break;

case 5:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die5, 0, 0, 160, 160);

ctx.restore();

break;

case 6:

ctx.save();

ctx.translate(w \* 0.84, h \* 0.7);

ctx.clearRect(0, 0, 160, 160);

ctx.drawImage(die6, 0, 0, 160, 160);

ctx.restore();

break;

}}

asyncForEach(die.roll(), chooseSide)