Sliding puzzle feladat

A kód egy klasszikus "tologatós játékot" (sliding puzzle) valósít meg, ahol számokat kell a helyükre mozgatni egy rácsban.



**A program felépítése**

A program három fő fájlból áll:

1. **index.html** - Az alap HTML struktúra
2. **style.css** - A megjelenés formázása
3. **script.js** - A játék fő logikája
4. **number.js** - A Number osztály definíciója

**HTML alapszerkezet (index.html)**

Az index.html egy egyszerű HTML5 dokumentum, amely:

* Beállítja a viewport-ot a reszponzív megjelenéshez
* Betölti a style.css stíluslapot
* A játék címe "Tologatós játék"
* A body elem üres, a rács dinamikusan jön létre JavaScript segítségével
* Betölti a script.js szkriptet modul típusként

**Stílus (style.css)**

A CSS fájl meghatározza:

* A rács konténer megjelenését (grid-container)
* A rács elemek stílusát (grid-item)
* A számok megjelenését (number)

**Number osztály (number.js)**

Ez az osztály felelős az egyes számok létrehozásáért és kezeléséért:

**Konstruktor(value, numbers, size)**

value: A szám értéke (1-től size\*size-1-ig)

numbers: Hivatkozás az összes számot tartalmazó tömbre

x és y: A szám pozíciója a rácsban (kezdetben undefined)

size: A rács mérete

**setPosition metódus**

1. Ha a számnak már volt korábbi pozíciója, kitörli onnan
2. Létrehoz egy új div elemet a számhoz
3. Elhelyezi a számot az új pozícióban
4. Frissíti a szám x és y koordinátáit

**createElement metódus**

1. Létrehoz egy új div elemet a számhoz
2. Beállítja a megfelelő osztályt és szövegét
3. Hozzáad egy kattintás eseménykezelőt, ami:
   * Megkeresi a mellette lévő üres pozíciót
   * Ha van ilyen, átmozgatja a számot az üres pozícióba

**emptyPosition getter**

Ez a getter ellenőrzi, hogy van-e üres pozíció a szám körül (balra, jobbra, fent, lent).

**checkIsEmpty metódus**

Ez a metódus ellenőrzi, hogy egy adott pozíció üres-e (nincs-e rajta szám).

## Fő játék logika (script.js)

### **Inicializálás**

* A rács mérete 10x10-es
* Létrehozza a rácsot
* Inicializál egy üres tömböt a számoknak
* Véletlenszerűen elhelyezi a számokat

**createGrid függvény**

* Létrehoz egy grid-container div-et
* Beállítja a grid oszlopainak számát a méret alapján
* Létrehoz size\*size darab grid-item div-et
* Beállítja a magasságukat és betűméretüket arányosan a mérethez
* Hozzáadja a grid-container-t a body-hoz

**randomizeNumbers függvény**

* Létrehoz *size-1 darab számot (1-től size*-1-ig)
* Minden számhoz véletlenszerű, még szabad pozíciót keres
* Elhelyezi a számot a talált pozícióban

**isFree függvény**

Ez a függvény ellenőrzi, hogy egy adott pozíció üres-e a DOM-ban.

## A játék működése lépésről lépésre

1. **Inicializálás**:
   * A böngésző betölti az oldalt
   * Létrejön egy 10x10-es rács
   * 99 szám (1-től 99-ig) véletlenszerűen elhelyezkedik a rácsban
   * 1 pozíció üres marad
2. **Játékmenet**:
   * A felhasználó a számokra kattint
   * Ha a szám mellett van üres hely (balra, jobbra, fent vagy lent), akkor:
     + A szám átmozog az üres helyre
     + Az eredeti helye üressé válik
   * Ha nincs üres hely mellette, semmi sem történik
3. **Győzelem feltétele**:
   * A kódban nincs explicit győzelmi feltétel implementálva
   * A játék klasszikus szabályai szerint akkor nyer a játékos, ha minden számot sorrendben helyez el

## Követelmények

* Fejlesztés nyelve: TypeScript
* Legyen a fejlesztés objektumorientált elven elkészítve.
* Készíts osztályt a számokhoz