

JavaScript évfolyam ZH

Due 2022. nov 4 by 19:00 **Pont** 30 **Beküldés...** egy fájlfeltöltés **Fájltípusok** zip
Elérhető 2022. nov 4, 16:00 - 2022. nov 4, 19:00 körülbelül 3 óra

Ez a feladat zárólva lett ekkor: 2022. nov 4, 19:00.

Tudnivalók

- A zárthelyi megoldására **180 perc** áll rendelkezésre, amely **magába foglalja** a kezdőcsomagban található `README.hun.md` fájl kitöltésére, a feladatok elolvasására, az anyagok letöltésére, összecsomagolására és feltöltésére szánt időt is.
- A feladatokat a **Canvas és Teams rendszeren keresztül is** be kell adni. **A rendszer pontban 19:00-kor lezár, ezután nincs lehetőség beadásra.**
- A feladatok megoldásához **bármilyen segédanyag használható** (dokumentáció, előadás, órai anyag, cheat sheet). A zh időtartamában igénybe vett **emberi segítség tilos** (szinkron, aszinkron, chat, fórum, stb)! Erről nyilatkoztok az alább olvasható `README.hun.md` fájlban is, ahol tudomásul veszitek ennek következményeit.
- A feladatok nem épülnek egymásra, **tetszőleges sorrendben** megoldhatók.
- A feladatok megoldásához először **töltsd le az általunk készített keretprogramot**. Ebben minden feladat külön könyvtárban helyezkedik el. Minden könyvtárban előkészítettük a HTML, CSS, JavaScript állományokat. Ezekben dolgozz! Általában csak a `.js` fájlhoz kell hozzányúlni, de ha kell, akkor a HTML is módosítható, sőt több `.js` fájlra is szétoszthatod a megoldásodat, de ez egyáltalán nem elvárás.
- A letöltött keretprogramban lévő `README.hun.md` fájlban töltsétek ki a nevetek és a Neptun azonosítókat (a `<>` jeleket nem kell beleírni)! **A megfelelően kitöltött `README.hun.md` fájl nélkül a megoldást nem fogadjuk el!**
- Minden feladat könyvtárban találsz egy `TASKS.md` fájlt. Ezekben az egyes `[]` közötti szöközt cseréld le `x`-re azokra a részfeladatokra, amiket sikerült (akár részben) megoldanod! Ez segít nekünk abban, hogy miket kell néznünk az értékeléshez.

1. feladat: Amőba (1-tictactoe, 8 pont)

Egy adott méretű rácson játszott amőba játék pillanatnyi állását tartalmazza a `game` konstans az `index.js` állományban. Oldd meg a következő feladatokat ezzel kapcsolatban! *Technikai segítség: A feladatok megoldása tömbfüggvényekkel ajánlott. Egy stringet tömbbé a `split` [.\(https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split\)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/split) függvény használatával alakíthatsz.*

- a. (2 pont) A `task1` azonosítójú elembe írd ki, hogy minden sor ugyanolyan hosszú-e! (Ugyanolyan-e hosszú minden sor, mint az első?)
- b. (2 pont) A `task2` azonosítójú elembe írd ki, hogy az első sorban csak X és O karakterek vannak-e!
- c. (2 pont) A `task3` azonosítójú elembe írd ki az X és O karakterek számát!
- d. (2 pont) A `task4` azonosítójú elembe add meg egy olyan sor indexét, amelyben három O vagy három X van egymás mellett! (Feltételezhető, hogy létezik legalább egy ilyen sor.)

TIC-TAC-TOE

Minden sor ugyanolyan hosszú? (true/false)

Do all rows have the same length? (true/false)

true

Az első sorban csak X vagy O karakter van? (true/false)

Does the first row only contain X and O characters? (true/false)

true

X és O karakterek száma:

Number of X and O characters:

X / O = 3 / 7

Egy olyan sor indexe, amelyben három O vagy három X van egymás mellett:

Index of any row that contains three consecutive Xs or three consecutive Os:

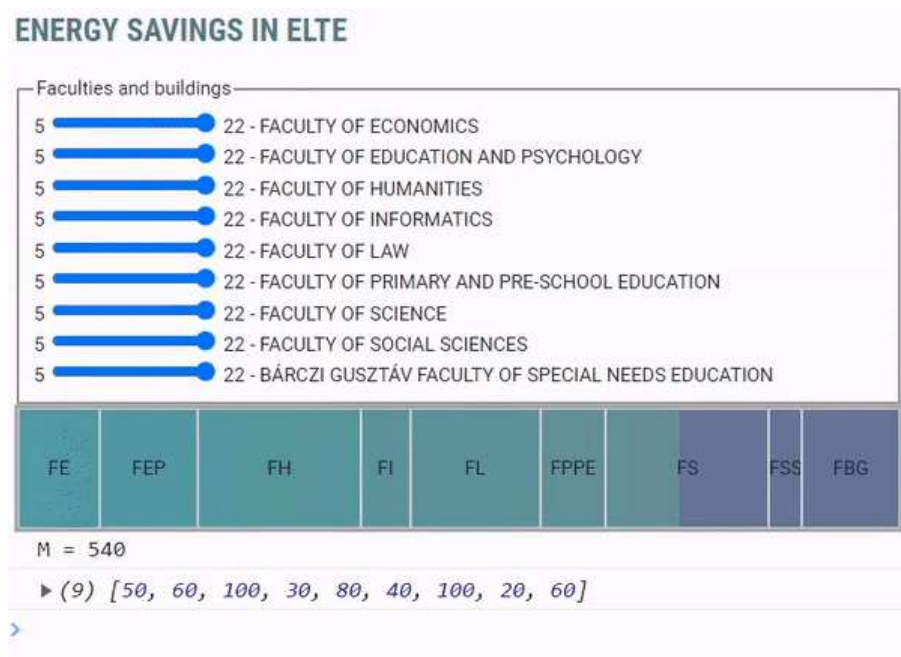
2

2. feladat: Energiamegtakarítás (2-savings, 8 pont)

Az elszálló energiaárak miatt a nagy épületekkel rendelkező szervezeteknek, mint az ELTÉnek különösen fontos, hogy minél több energiát takarítsanak meg! Ezt legkönnyebben a hőmérséklet alacsonyan tartásával lehet elérni. Segíts az ELTÉnek 25%-os

költségmegtakarítást elérni azáltal, hogy beállítjuk az egyes épületek hőmérsékletét! A hőmérsékleteket egy-egy csúszkával lehet beállítani, 5 és 22 fok között. Minden épület esetében adott, hogy tavaly mennyit fogyasztott. Ez az inputok `data-consumption` attribútumában van megadva.

- a. (2 pont) Számold ki a tavalyi év össz fogyasztását (`M`) és írd ki a konzolra! Ehhez vedd az összes inputot, olvasd ki a `data-consumption` értékeket számként, majd add össze!
- b. (2 pont) Minden input esetén számold ki az aktuális fogyasztását (`ci`) és írd ki a konzolra! Ehhez olvasd ki az input aktuális értékét (`value`), oszd el a maximális értékkel (`max` attribútuma az inputnak) és szorozd meg a `data-consumption` értékével! (`ci = (value/max)*consumption`) Megjegyzés: az oldal betöltésekor a `ci` értéke megegyezik a `data-consumption` értékével, mivel mindegyik csúszka maximumon áll.
- c. (2 pont) Minden inputhoz tartozik egy `label` elem az űrlap alatti diagramon. A label elem `for` attribútuma mutatja, hogy melyik `id`-jű inputhoz tartozik. Minden input esetén állítsd be a hozzá tartozó `label` elem szélesség stílus tulajdonságát `ci / M * 100` százalékra (pl. `65%`)!
- d. (2 pont) Ha bármely csúszka értékét változtatjuk (`input` esemény), akkor aktualizáljuk a diagramot! (Azaz az első három pontot újra és újra hajtsuk végre!) *Technikai segítség: elég az űrlap szintjén figyelni az `input` eseményeket, és akkor újraszámolni az értékeket.*



3. feladat: Csomagkiszállítás (3-parcel, 8 pont)

Távoli helyszínekre repülőgéppel szállítják a csomagokat. A leszállás és felszállás sok üzemanyagot emészt fel, az ügyes pilóta röptében is képes pontosan kézbesíteni a csomagot! Készíts egy canvas-es játékot, ahol az első kattintásra elindul a repülőgép, a másodikra pedig elengedi a csomagot! Legyél ügyes, találj célba!

Az `index.js` fájl már tartalmazza a játékhoz szükséges kereteket:

- a `next` játékciklust - ahogy az [előadáson](http://webprogramozas.inf.elte.hu/webprog/lectures/06/#/anim%C3%A1ci%C3%B3s-ciklus-1) [_\(\[http://webprogramozas.inf.elte.hu/webprog/lectures/06/#/anim%C3%A1ci%C3%B3s-ciklus-1\]\)](http://webprogramozas.inf.elte.hu/webprog/lectures/06/#/anim%C3%A1ci%C3%B3s-ciklus-1) is volt
- az állapottér módosításához és kirajzolásához szükséges függvényeket (`update` és `render`)
- az állapotteret a játékhoz szükséges objektumok adataival: repülő (`plane`), csomag (`parcel`) és ház (`house`)

A feladataid:

- a. (1 pont) A `render` függvényen belül töröld le a vásznat, majd rajzolj ki 3 téglalapot a repülő, csomag és ház helyére!
- b. (1 pont) A téglalapok helyett az előre betöltött képeket jelenítsd meg!
- c. (2 pont) Kattintásra induljon el a repülő! Ehhez a repülő vízszintes sebességét (`vx`, pl. 200px/s) kell beállítani, és természetesen ennek megfelelően változtatni a pozícióját!
- d. (1 pont) A repülővel együtt mozogjon a csomag is!
- e. (1 pont) Újabb kattintásra ejtsd le a csomagot! Ehhez a csomagnak kell egy függőleges irányú gyorsulást adni (`ay`, pl. 300 px/s²). Használhatod a `gameState` változót a megfelelő állapot nyomon követésére.
- f. (1 pont) Minden elmozduláskor ellenőrizd, hogy a csomag találkozik-e a házzal! Ha igen, akkor jeleníts meg egy szöveget a rajzvászonon ennek megfelelően ("Delivered"), és állítsd meg a csomag zuhanását!

- g. (1 pont) Ha a csomag eléri a vászon alját és nem találkozott a házzal, akkor ezt is jelenítsd meg egy szöveggel a rajzvászonon ("Missed"), és állítsd meg a csomag zuhanását!

PARCEL DELIVERY



4. feladat: Memóriajáték (4-memory, 8 pont)

Írj egy olyan játékot, ahol felvillanó körök sorrendjét kell megjegyezni, majd emlékezetből visszajátszani!

- a. (2 pont) A körök számaért felelős csúszkát mozgatva, rajzolj ki a csúszka értékének megfelelő számú kört a `container` azonosítójú elembe! Az `index.html`-ben találsz példát arra, hogy hogyan néz ki egy kör.
- b. (1 pont) A "Start" gomb megnyomására töltsd fel a `series` tömböt 7 darab, 1 és a körök száma közötti (határokat beleértve) véletlen egész számmal! Írd is ki a konzolra a `series` tömb értékét!
- c. (1 pont) Ugyancsak a "Start" gombra kattintva villantsd fel az első kört! Ehhez használd a `toggleHighlight` segédfüggvényt, paraméterként megadva az adott kör DOM elemét! A segédfüggvény egy animációt ad a kiválasztott elemhez, majd el is távolítja azt.
- d. (1 pont) Ha egy kört fel tudsz villantani, akkor menj végig a generált `series` tömbön, és villantsd fel a tömb eleminek megfelelő sorszámú köröket! Ha ezt ciklussal csinálnánk, akkor mindegyik kör egyszerre villana fel. Menjünk végig a tömbön 1 másodperces késleltetéssel! Ehhez használd az előadáson mutatott ötletet: a számlálós ciklust írd át előltesztelésre, majd azt rekurzióra, végül a rekurzív hívást helyettesítsd időzítéssel!

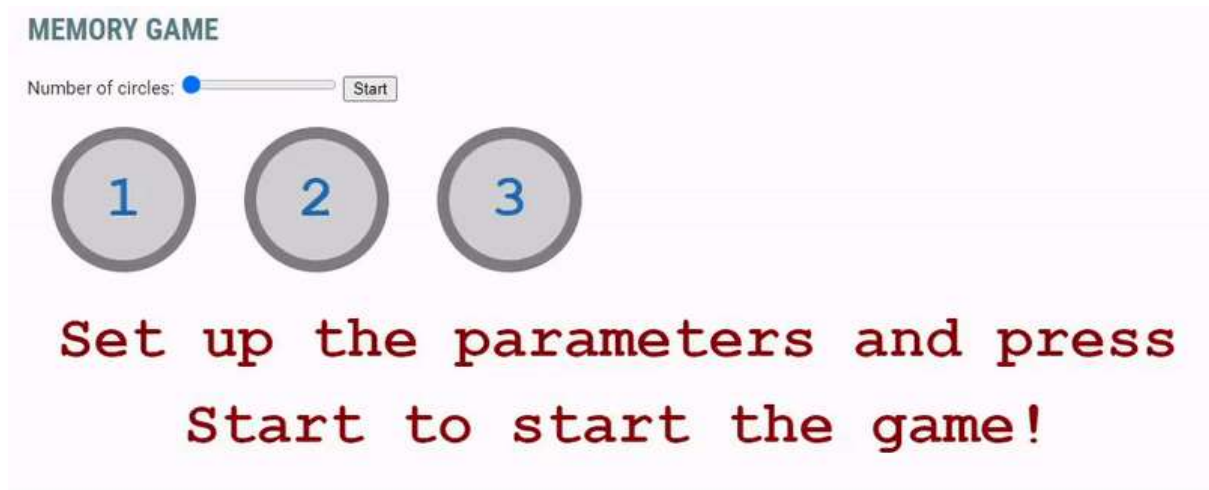
```
function foo() {
  for (let i = 0; i < 5; i++) {
    // something
  }
}

function foo(i = 0) {
  while (i < 5) {
    // something
    i = i + 1
  }
}

function foo(i = 0) {
  if (i < 5) {
    // something
    foo(i + 1)
  }
}

function foo(i = 0) {
  if (i < 5) {
    // something
    setTimeout(() => foo(i + 1), 1000)
  }
}
```

- e. (1 pont) A körökre kattintva olvasd ki a kattintott kör számát, majd rakd bele a `solution` tömbbe! Minden ilyen alkalommal ellenőrizd, hogy a `solution` tömb hossza elérte-e a `series` tömb hosszát, és ha igen, akkor megegyeznek-e! Az eredményt írd ki a konzolra!
- f. (1 pont) Az `output` azonosítójú elembe a körök villantásakor írd ki, hogy "Flashing circles...", majd az összes villanás befejeződése után a "Now, your turn..." szöveget! A játék végén írd ugyanebbe az elembe a "Nice job!" vagy a "Failed!" szöveget az eredménytől függően!
- g. (1 pont) Egy állapotváltozó (`canGuess`) segítségével ne engedjünk addig tippelni, amíg a gép fel nem villantotta a megfelelő darabszámú kört!



Kritérium	Értékelések		pontok
1/a.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
1/b.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
1/c.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
1/d.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
2/a.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
2/b.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
2/c.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
2/d.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
3/a.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
3/b.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
3/c.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont
3/d.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
3/e.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
3/f.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
3/g.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/a.	2.0 pont Teljes megoldás	0.0 pont Nem értékelhető	2.0 pont

Kritérium	Értékelések		pontok
4/b.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/c.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/d.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/e.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/f.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
4/g.	1.0 pont Teljes megoldás	0.0 pont Nem értékelhető	1.0 pont
Összpontszám: 32.0 pontból			