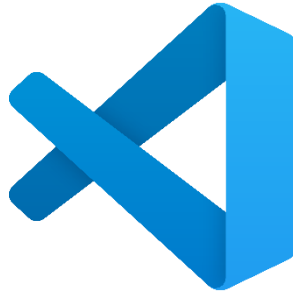




Projektfeladat: Amőba játék

Készítette: Kanizsai Ferenc

2026.01.21. – 2026.02.04.



A projekt célja

Célja, hogy elsajátítsam a HTML, CSS és JavaScript alapjait, hogyan lehet felépíteni szerkezetét, hogyan lehet stílusosan és reszponzívan megjeleníteni és hogy lehet dinamikussá tenni egy weboldalt.

A projektben használt hardveres eszközök

PC – Intel i5-10400f - 16GB RAM - 1TB SSD [**Kliens**]

Internetkapcsolat – dokumentációk, oktatóanyagok eléréséhez [**Egyéb**]

A projektben használt szoftveres eszközök

Visual Studio Code [**Kódszerkesztő**]

HTML [**Programozási nyelv**]

CSS [**Programozási nyelv**]

JavaScript [**Programozási nyelv**]



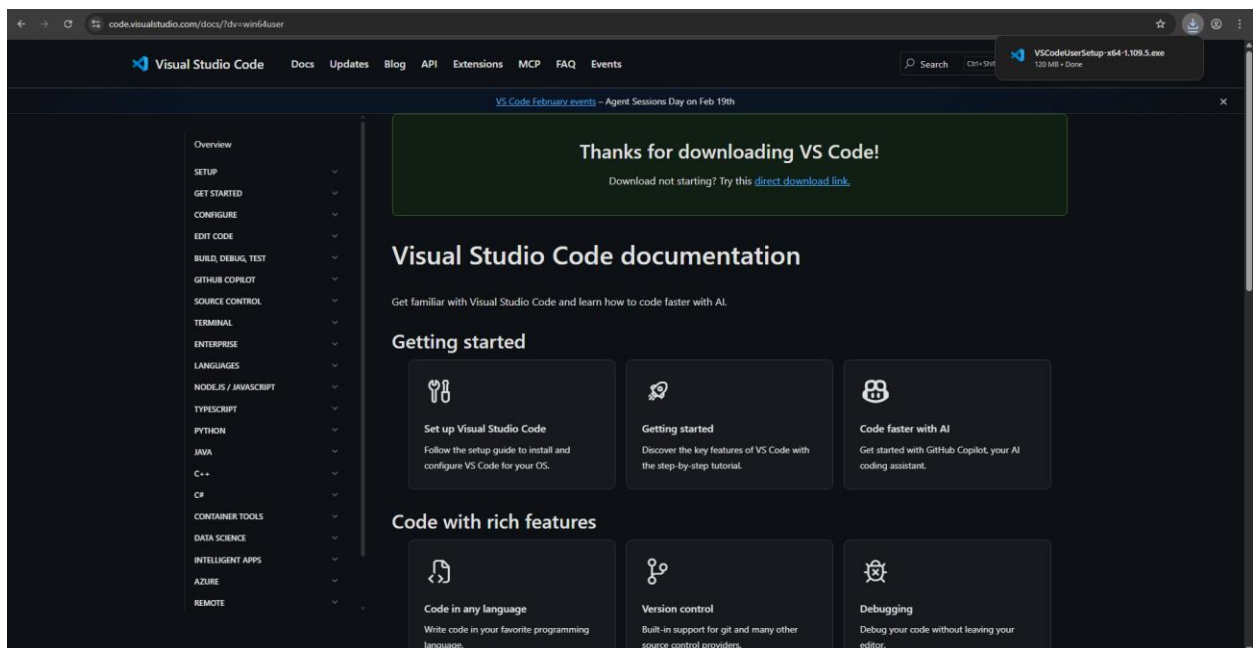
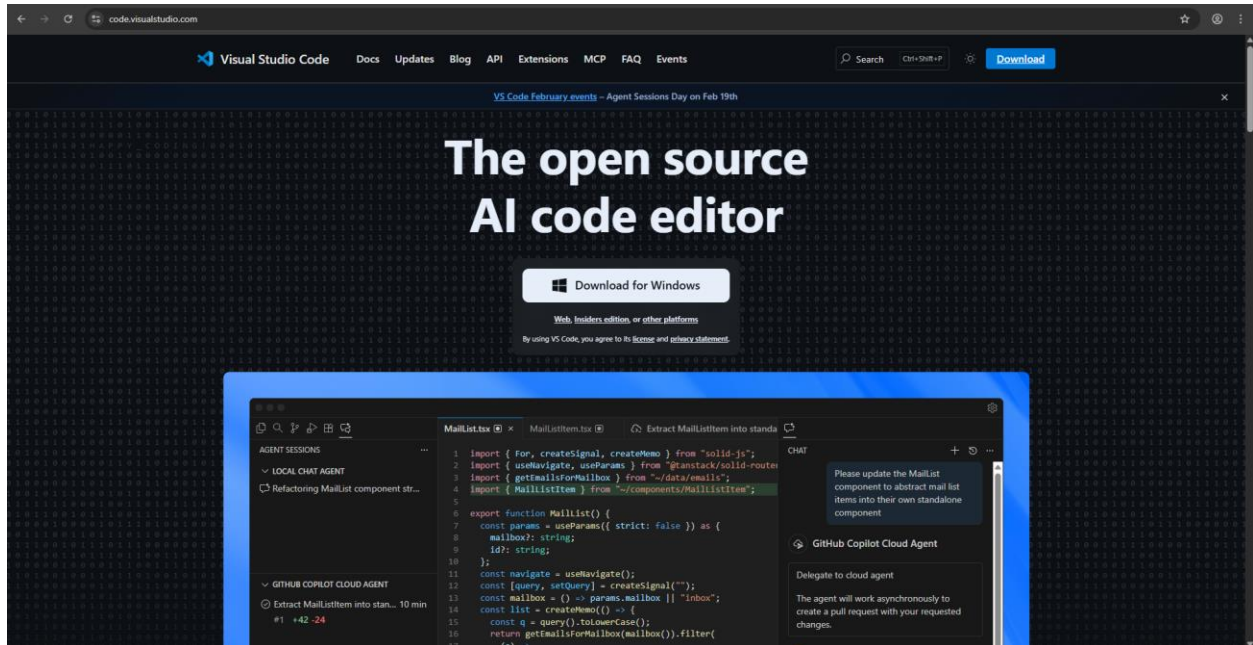


KECSKEMÉTI SZAKKÉPZÉSI CENTRUM KANDÓ KÁLMÁN TECHNIKUM

6000 Kecskemét, Bethlen krt. 63., Telefon: 76/481-622
Fax: 76/485-971, E-mail: kando@kecskemetiszc.hu, Web: www.kkando.hu

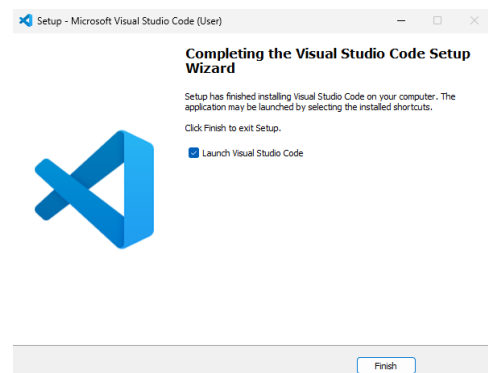
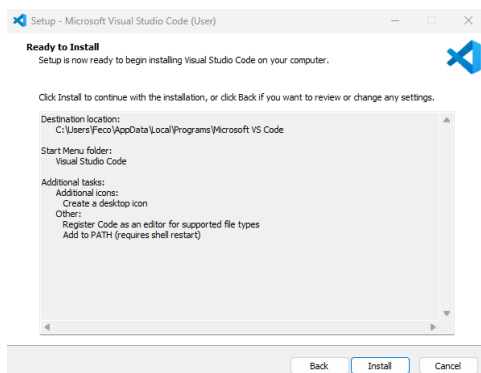
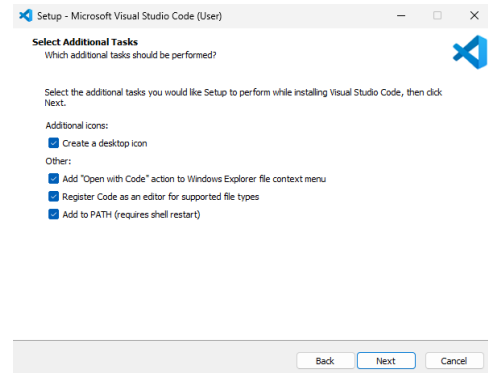
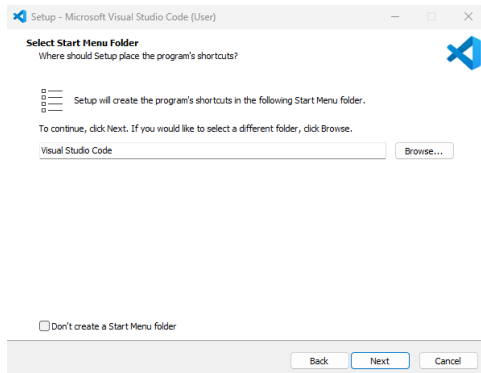
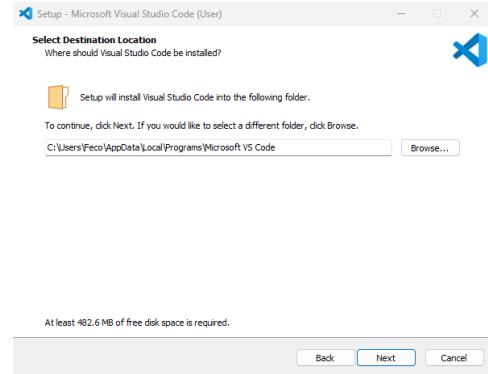
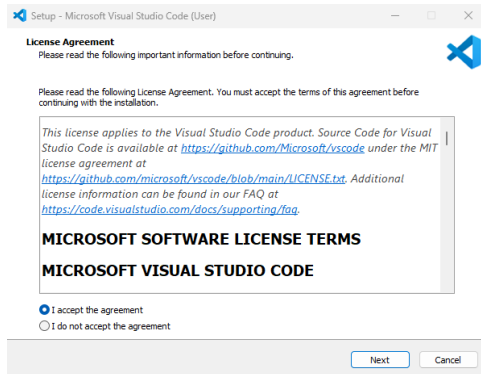
Visual Studio Code telepítése

- Első lépésként töltsük le az kódszerkesztőt: <https://code.visualstudio.com/>





- Telepítsük a Visual Studio Code-t

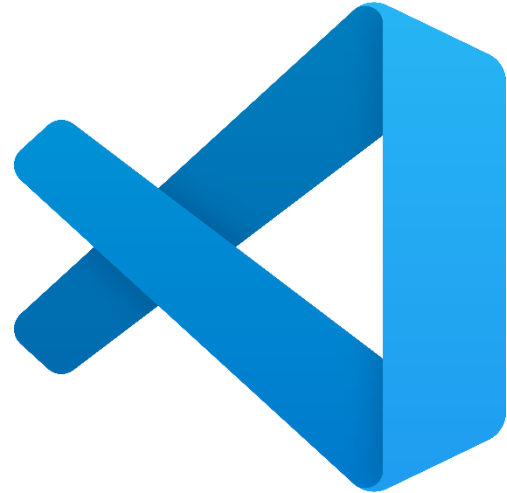




Mi az a Visual Studio Code?

A **Visual Studio Code (VS Code)** egy ingyenes, nyílt forráskódú forráskód-szerkesztő, amelyet a **Microsoft** fejleszt.

Lehetővé teszi, hogy különböző programozási nyelveken írjunk, szerkesszünk és futtassunk kódot egy modern, gyors és testre szabható környezetben.



Mikor jelent meg?

Az első verziója 2015-ben jelent meg, azóta folyamatosan frissítik, fejlesztik, új funkciókat kap.

Hol használják?

IT cégeknél, egyetemeken és iskolákban, használják még programozók, webfejlesztők, rendszergazdák és hobbi fejlesztők.

Mire lehet használni?

Weboldalak készítése, játékfejlesztés, Python, Java, C++, JavaScript programok írására, jó tanuláshoz, gyakorláshoz, illetve csapatmunkához is jó.

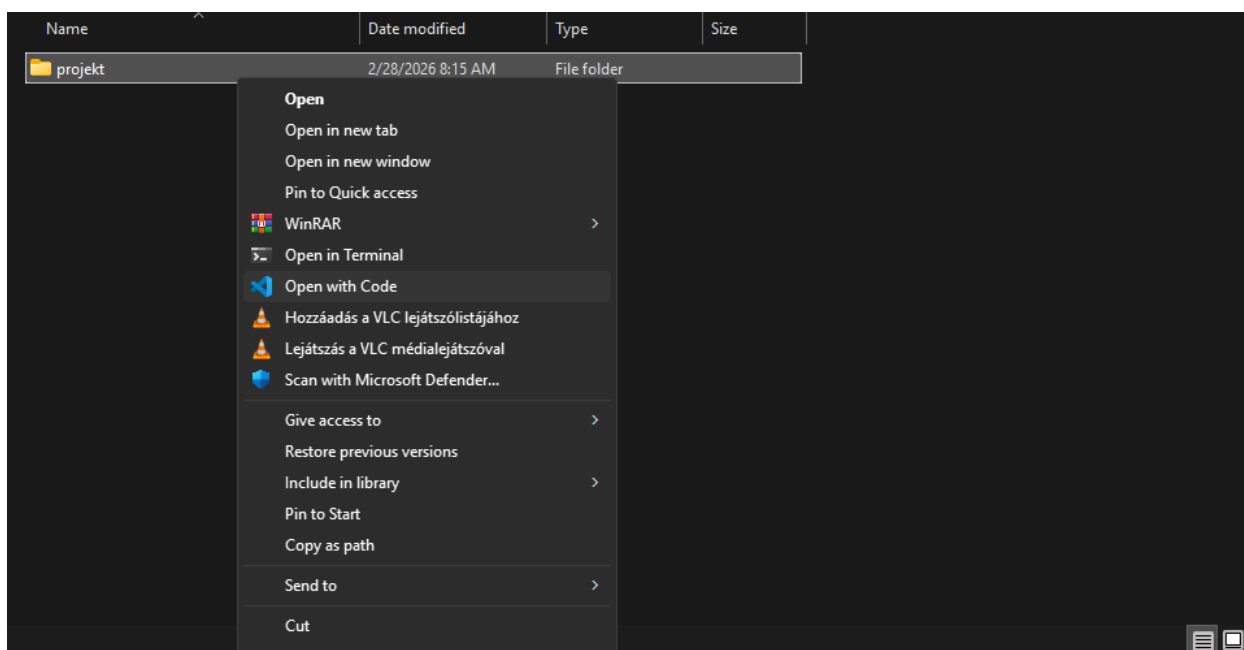
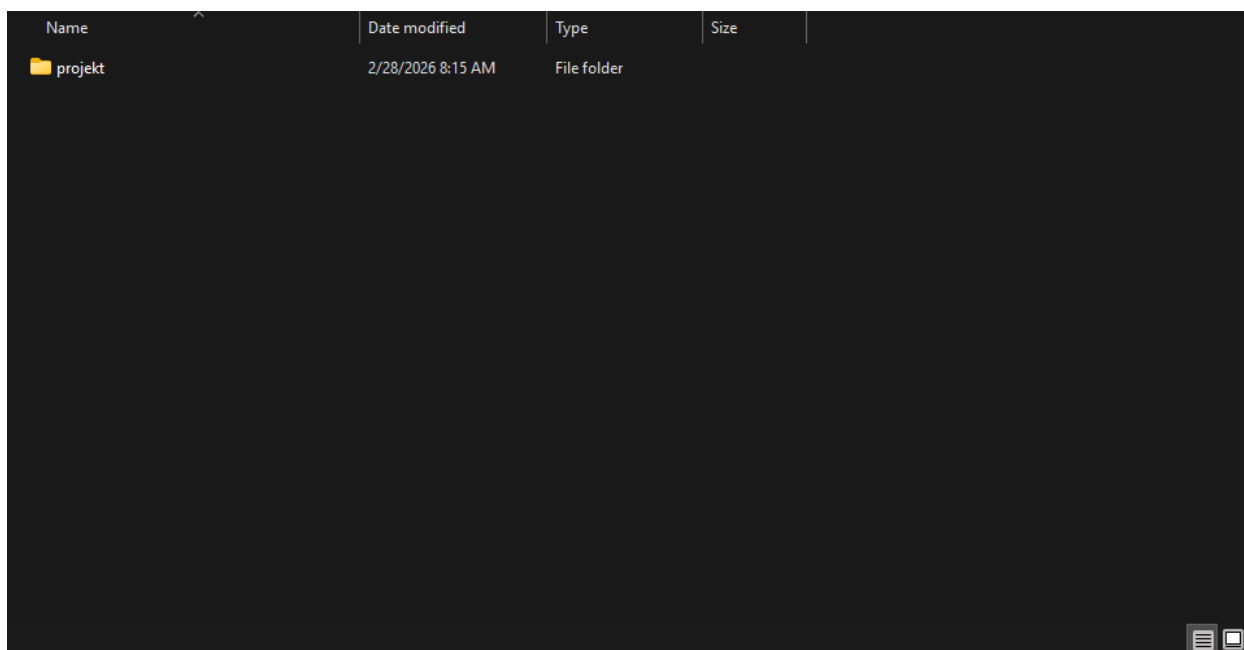
Összefoglalva:

Ez a program egy modern, gyors kódszerkesztő, amivel hatékonyan lehet programozni.



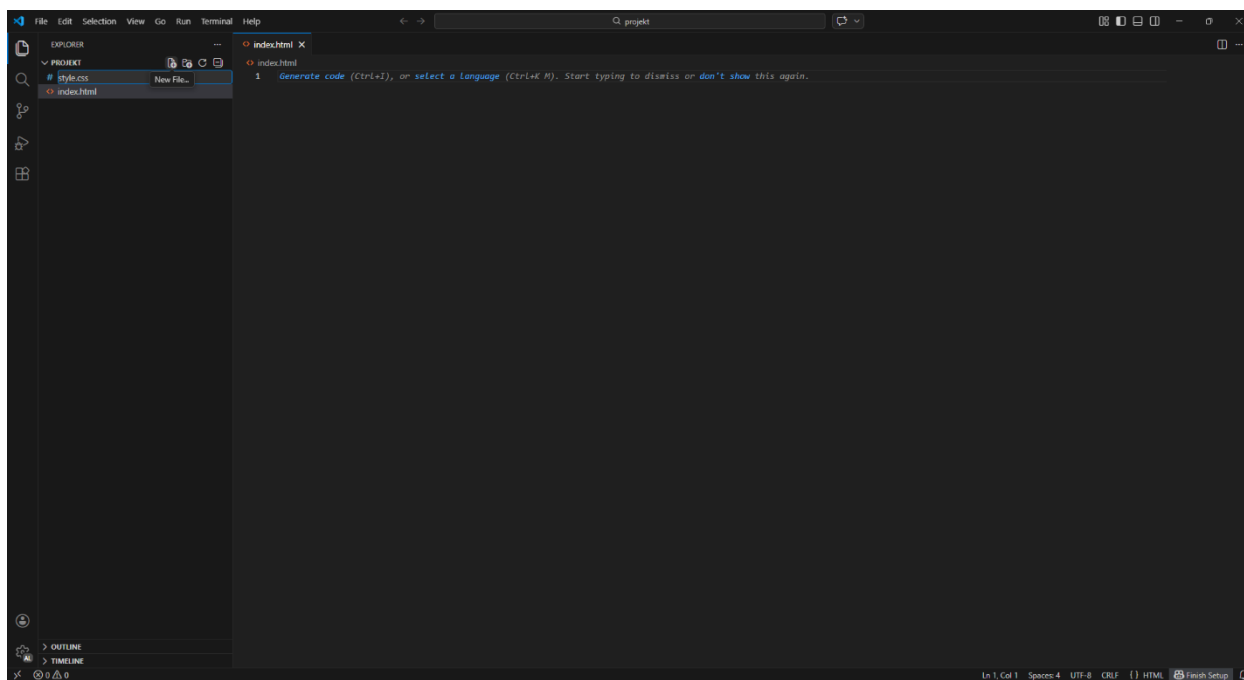
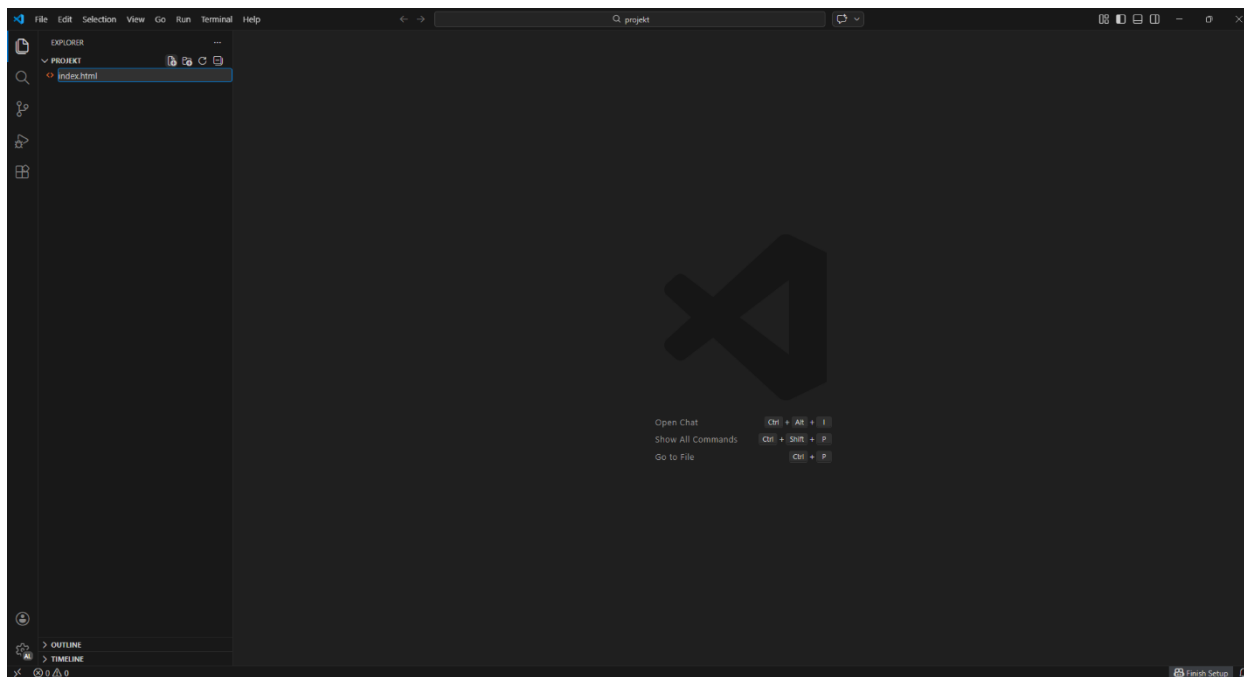
Visual Studio Code használata

- Hozzunk létre egy mappát, majd jobb kattintással nyissuk meg a mappát Visual Studio Code-ban:





- Hozzunk létre fájlokat, én index.html és style.css fájlokat fogok létrehozni:





- A VS Code egy egyszerű programozási környezetet teremt nekünk. Ha beírunk egy „!” jelet, a program már felajánlja az alap sablont a html fájlnak.

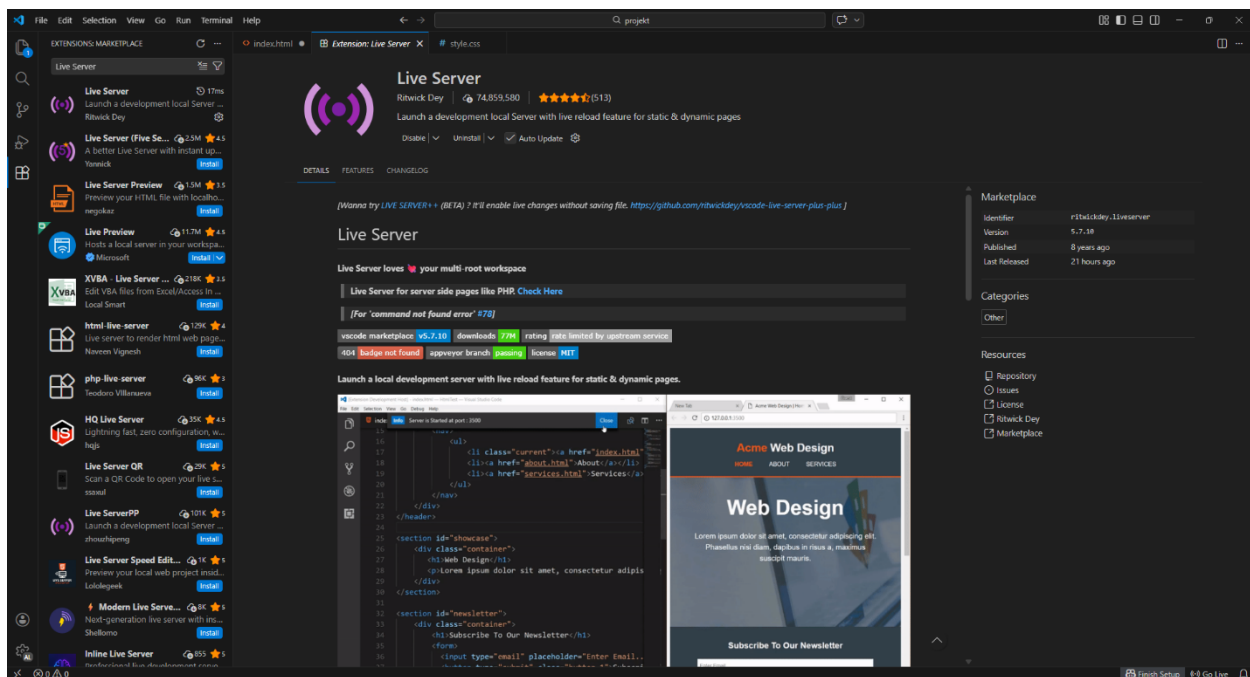
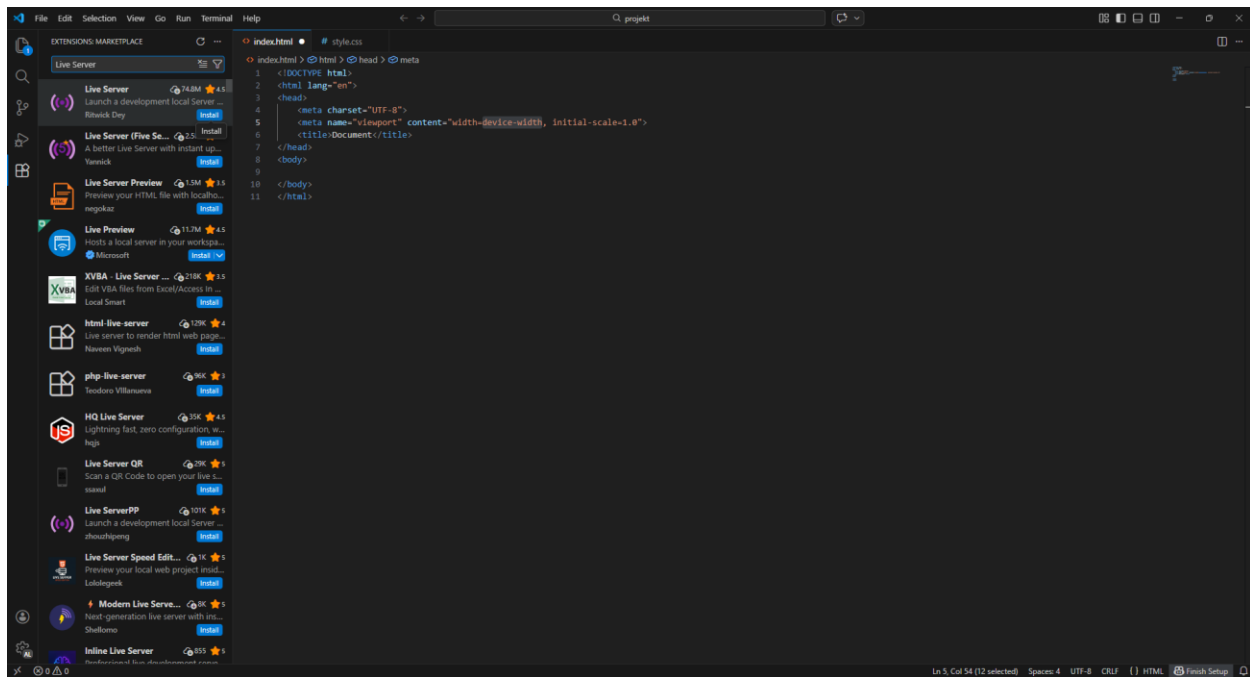
```
1 |  
  !!!
```

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6   <title>Document</title>  
7 </head>  
8 <body>  
9  
10 </body>  
11 </html>
```



Visual Studio Code Live Server telepítése

- Menjünk az „Extensions” részhez, majd írjuk be a keresőbe, hogy „Live Server”, aztán nyomjunk az install gombra.





Mi az a Live Server?

A **Live Server** egy bővítmény (extension) a **Visual Studio Code** számára, amely lehetővé teszi, hogy a HTML, CSS és JavaScript fájlokat **valós időben, automatikus frissítéssel** jelenítsük meg a böngészőben.



Mikor jelent meg?

2017 körül jelent meg a Visual Studio Code bővítmények között, Ritwick Dey készítette.

Hol használják?

Webfejlesztés.

Mit csinál?

Indít egy helyi webszervert a gépen, majd megnyitja a weboldalt a böngészőben, nagy előnye, hogy minden mentésnél azonnal frissíti a weboldalt, ezáltal valós előnézetet ad a kódhoz.

Mire lehet használni?

Weboldalak tesztelésére, hibakeresésre, gyors prototípus készítésére.

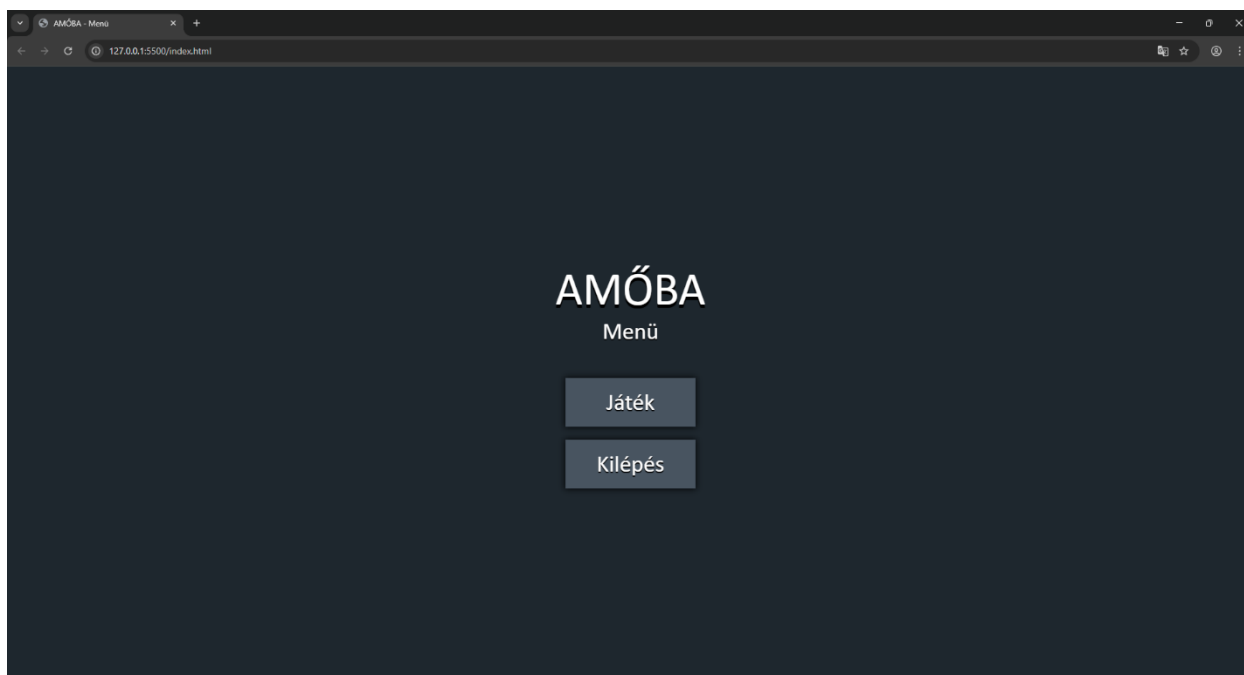
Összefoglalva:

Egyszerű, de nagyon hasznos eszköz, ami felgyorsítja a webfejlesztést.

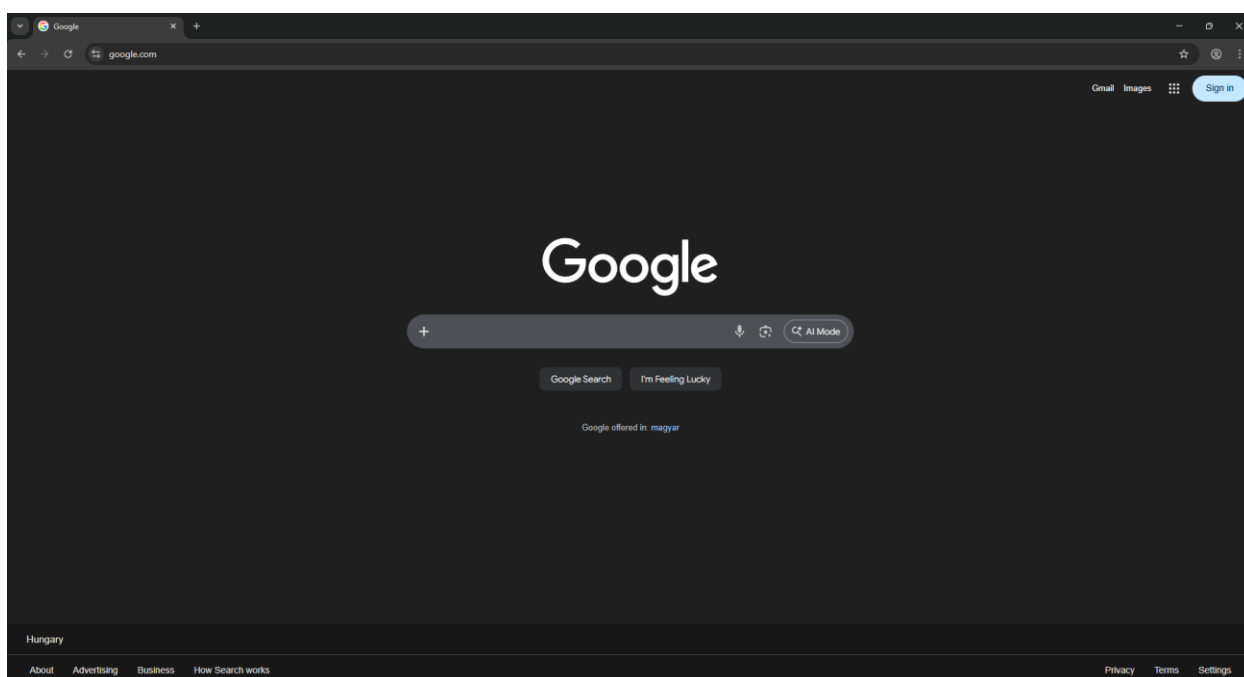


Amőba játék működése

- A játék első megnyitásakor a főmenübe érkezünk, ahol el tudjuk dönteni, hogy játszani akarunk, vagy kilépünk.

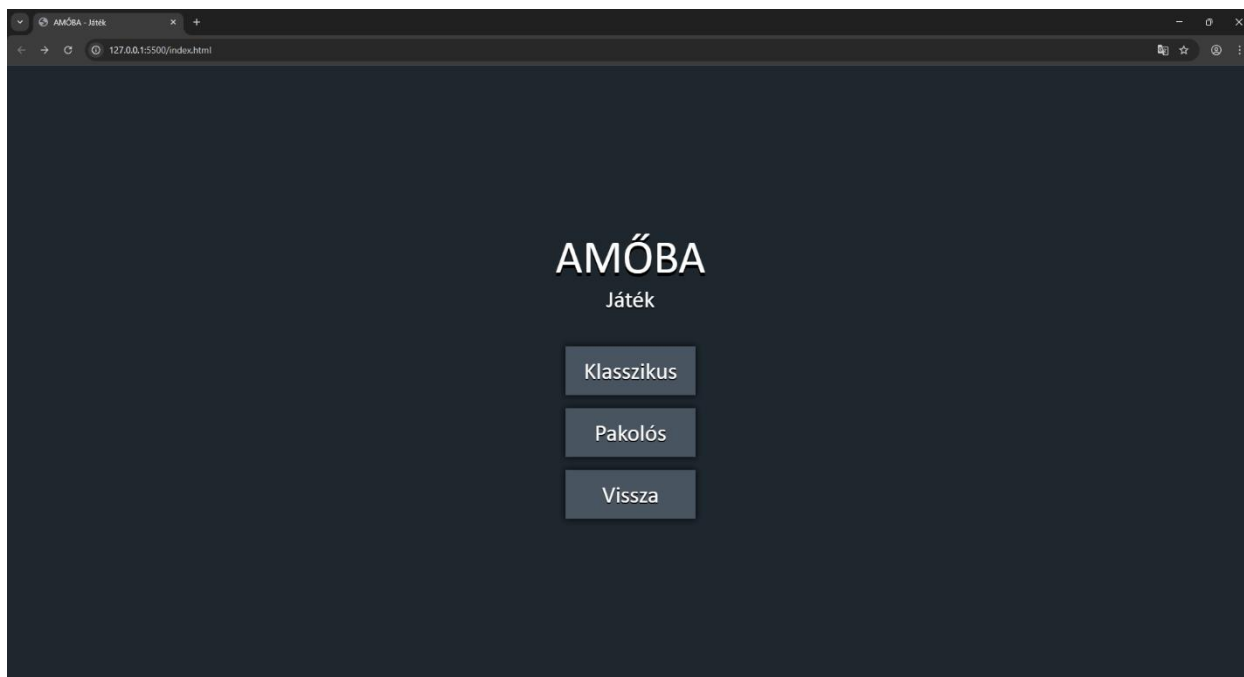


- Kilépés: Kidob minket a google.com oldalra.

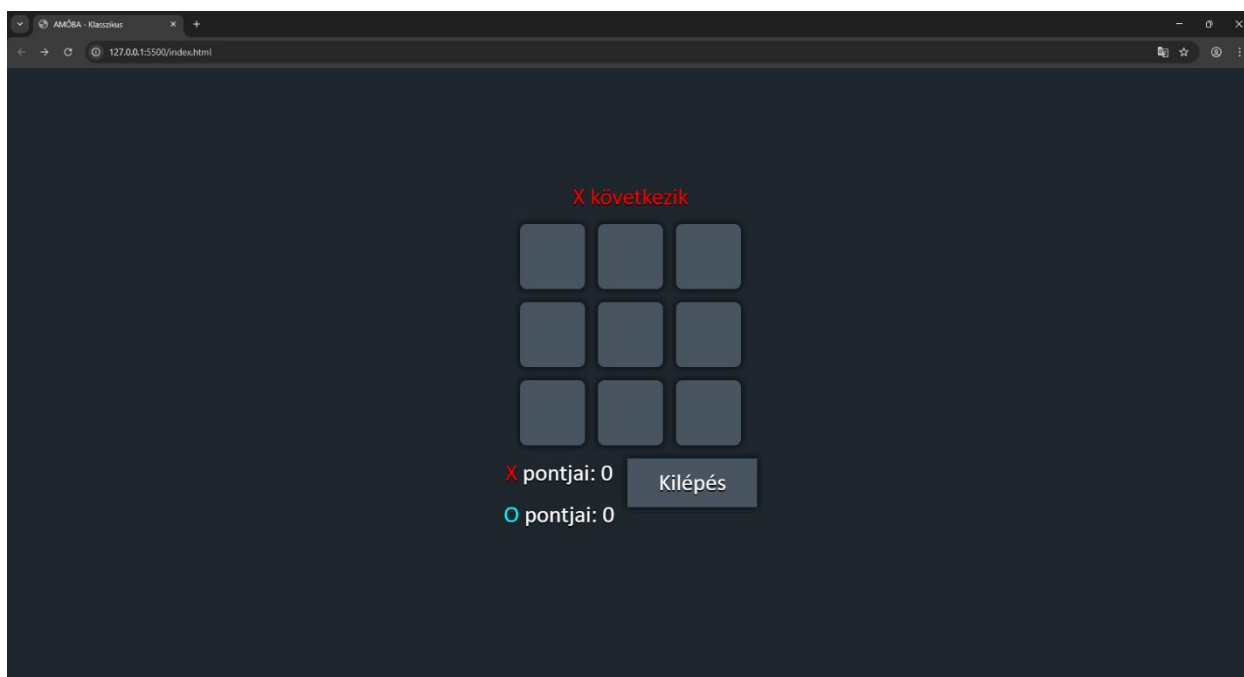


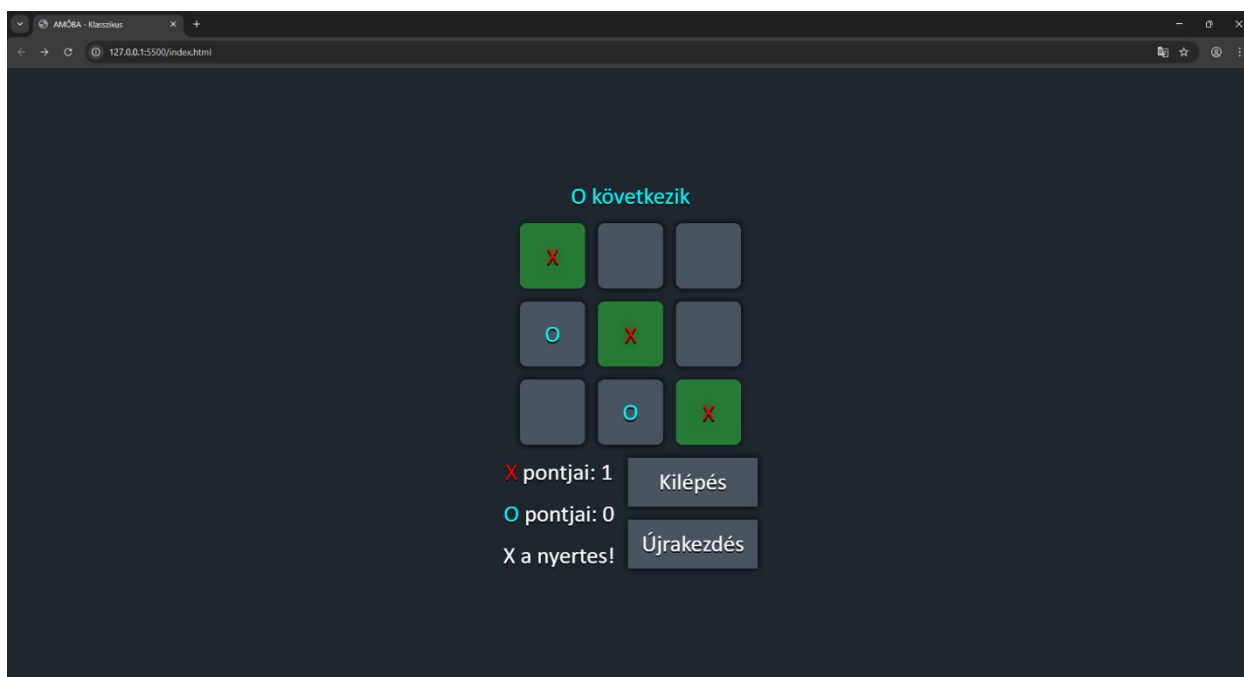


- Játék: Bedob minket a választó menübe, ahol 2 játékmód közül választhatunk (Klasszikus / Pakolós), illetve vissza tudunk lépni a főmenübe.

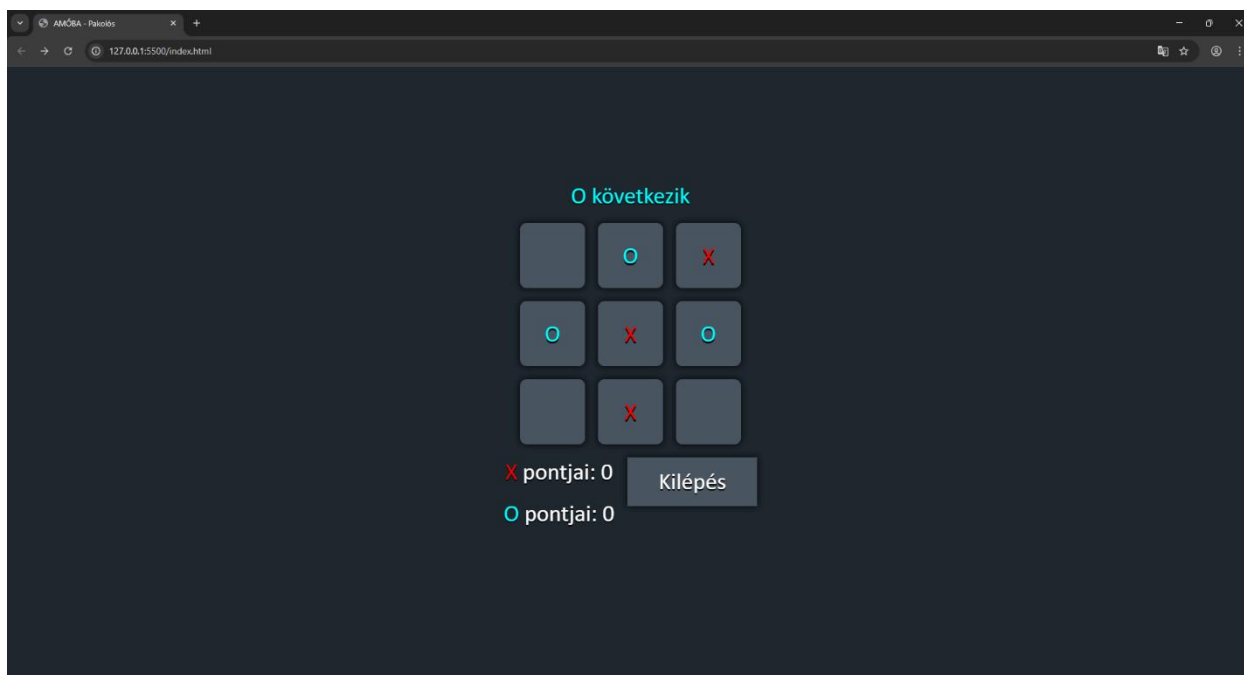


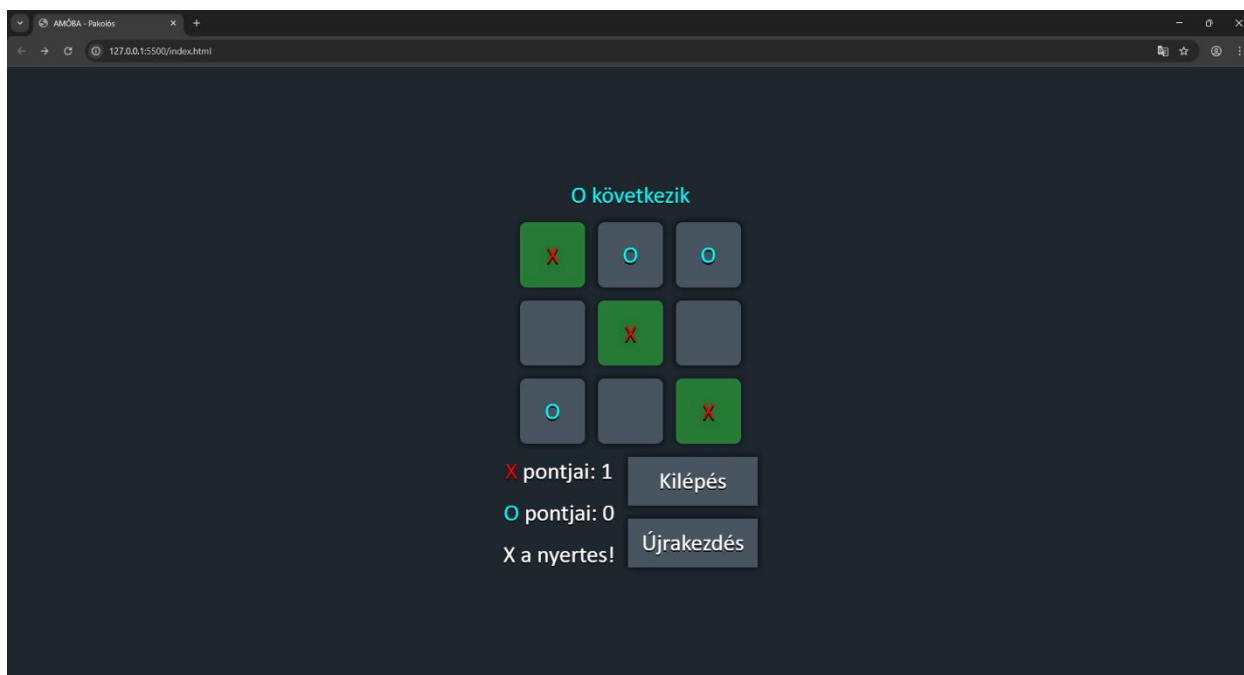
- Klasszikus játékmód: 3x3-as tábla, X és O szimbólumok vannak, akinek sikerül 3 azonos szimbólumot kirakni vízszintesen, függőlegesen, vagy keresztbe, az nyer.





- Pakolás játékmód: Itt az X-nek és a O-nek maximum 3-3 szimbóluma lehet a pályán. Ha letették mindet, akkor elkezdődik a pakolgatás.





- A rendszer számolja a pontokat, ezáltal végig lehet követni, hogy ki áll nyerésre.
- Ha valakinek kigyűlik a szimbólum, a rendszer átállítja a megfelelő háttereket zöldre, illetve nem lehet többet pakolni, kiírja, hogy ki nyert.
- Ha szeretnénk még játszani, az Újrakezdés gombra kattintva ezt megtehetjük.
- Ha kisseretnénk lépni a játékból, esetleg játékmódot váltani, akkor azt a kilépés gombbal tehetjük meg.



Amőba játék elkészítése HTML, CSS, JavaScript segítségével

- A játék felépítése:

```
# global.css
<> index.html
JS index.js
# page_JATEK.css
# page_KLASSZIKUS.css
# page_MENU.css
# page_PAKOLOS.css
```

- global.css: globális stílus, ami minden oldalra érvényes.

- index.html: ez építi fel az oldalt strukturálisan.

- index.js: ez teszi az oldalt dinamikussá, logikák felépítésére jó.

- page_(név): ezek a különböző oldalak stílusára vonatkoznak



- index.html fájl:

```
1 <!doctype html>
2 <html lang="hu">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <link rel="stylesheet" href="global.css" />
7     <link id="page_css" rel="stylesheet" href="" />
8     <title id="page_title"></title>
9   </head>
10  <body>
11    <script src="index.js"></script>
12  </body>
13 </html>
```

- Az oldal JavaScript segítségével épül fel, ezért nincs semmi tartalma a HTML fájlnek.

- Megvannak határozva bizonyos elemeknek ID-k, ez azért kell, hogy a javascript fájlban tudjunk rájuk hivatkozni és ezáltal áttudjuk írni az értéküket.

- A global.css-hez nincs id, mivel az minden oldalára kötelezően rákell, hogy kerüljön.



- global.css és egyéb css fájlok:

```
1 * {
2   margin: 0;
3   padding: 0;
4
5   user-select: none;
6   -webkit-user-drag: none;
7
8   box-sizing: border-box;
9
10  font-family: Calibri;
11  color: white;
12 }
```

- A .css fájlok az oldal stílusát határozzák meg, hogyan nézzen ki az oldal, milyen színekben tűnjön fel.

- Ami itt fontos, hogy minden oldalnak alaptól Calibri betűtípus van megadva és fehér szín, illetve nem lehet kijelölni szövegeket.



- index.js fájl:

```
1 function setCSS(fajl) {  
2     const page_css = document.getElementById("page_css");  
3     page_css.href = fajl;  
4 }  
5  
6 function setTitle(title) {  
7     const page_title = document.getElementById("page_title");  
8     page_title.textContent = title;  
9 }
```

- 2 funkció, amik szinte mindegyik load funkcióban benne vannak.

- Lényegük: A setCSS-el betudjuk az adott oldalnak a stílusát, a setTitle-val pedig az oldal címét.



```
1 function loadMenu() {
2   setCSS("page_MENU.css");
3   setTitle("AMŐBA - Menü");
4
5   document.body.innerHTML = `
6     <div id="FELSO-RESZ">
7       <div id="CIM"><span>AMŐBA</span></div>
8       <div id="ALCIM"><span>Menü</span></div>
9     </div>
10    <div id="ALSO-RESZ">
11      <div id="JATEK"><span>Játék</span></div>
12      <div id="KILEPES"><span>Kilépés</span></div>
13    </div>
14  `;
15
16  document.getElementById("JATEK").addEventListener("click", function () {
17    loadJatek();
18  });
19
20  document.getElementById("KILEPES").addEventListener("click", function () {
21    window.location.href = "https://www.google.com";
22  });
23 }
```

- load funkciók, ezek töltik be az oldalak struktúráját.

- Az elején beállítjuk az oldalnak a stílusát és címét, majd pedig a document.body.innerHTML-el beállítjuk a html fájl struktúráját, hogyan rendeződjenek el bizonyos elemek a weblapon.

- Az addEventListenerek alul gombok, amiknek kattintásakor végbemegy az a cselekedet, ami megvan neki adva.



- Amőba játék felépítése:

```
1 let pontok_x = 0;
2 let pontok_o = 0;
3
4 const KI_KOVETKEZIK_SZOVEG = document.getElementById("KI_KOVETKEZIK_SZOVEG");
5
6 const db1 = document.getElementById("db1");
7 const db2 = document.getElementById("db2");
8 const db3 = document.getElementById("db3");
9 const db4 = document.getElementById("db4");
10 const db5 = document.getElementById("db5");
11 const db6 = document.getElementById("db6");
12 const db7 = document.getElementById("db7");
13 const db8 = document.getElementById("db8");
14 const db9 = document.getElementById("db9");
15
16 const GYOZTES = document.getElementById("GYOZTES");
17
18 const KILEPES = document.getElementById("KILEPES");
19 const UJRAKEZDES = document.getElementById("UJRAKEZDES");
20
21 const szimbolum_x = "X";
22 const szimbolum_o = "O";
23
24 const pontok_x_szoveg = document.getElementById("pontok_x_szoveg");
25 const pontok_o_szoveg = document.getElementById("pontok_o_szoveg");
26
27 let jatekVege = false;
28
29 let szimbolum = szimbolum_x;
30
31 KI_KOVETKEZIK_SZOVEG.textContent = `${szimbolum} következík`;
32 KI_KOVETKEZIK_SZOVEG.style.color = "red";
```

- Meghatározott változók, a constansok azok nem változnak, a let változók már igen.



```
1 function checkSzimbolum() {  
2     if (szimbolum === szimbolum_x) {  
3         szimbolum = szimbolum_o;  
4         KI_KOVETKEZIK_SZOVEG.style.color = "aqua";  
5     } else {  
6         szimbolum = szimbolum_x;  
7         KI_KOVETKEZIK_SZOVEG.style.color = "red";  
8     }  
9     KI_KOVETKEZIK_SZOVEG.textContent = `${szimbolum} következik`;  
10 }
```

```
1 function checkPontok() {  
2     if (pontok_x > 0) {  
3         pontok_x_szoveg.textContent = `${pontok_x}`;  
4     }  
5     if (pontok_o > 0) {  
6         pontok_o_szoveg.textContent = `${pontok_o}`;  
7     }  
8 }
```



```
1 function checkWin() {
2     // 1. SOR
3     if (
4         db1.textContent === szimbolum_x &&
5         db2.textContent === szimbolum_x &&
6         db3.textContent === szimbolum_x
7     ) {
8         GYOZTES.textContent = `${szimbolum_x} a nyertes!`;
9         jatekVege = true;
10        pontok_x++;
11        db1.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
12        db2.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
13        db3.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
14    } else if (
15        db1.textContent === szimbolum_o &&
16        db2.textContent === szimbolum_o &&
17        db3.textContent === szimbolum_o
18    ) {
19        GYOZTES.textContent = `${szimbolum_o} a nyertes!`;
20        jatekVege = true;
21        pontok_o++;
22        db1.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
23        db2.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
24        db3.style.backgroundColor = "rgba(59, 224, 81, 0.5)";
25    }
```

- Ezek a check funkciók ellenőriznek, amikor kell.



```
1 db1.addEventListener("click", function () {
2   if (db1.textContent === "" && jatekVege === false) {
3     db1.textContent = szimbolum;
4     checkSzimbolum();
5     checkWin();
6     db1.style.cursor = "default";
7
8     if (db1.textContent === szimbolum_x) {
9       db1.style.color = "red";
10    } else if (db1.textContent === szimbolum_o) {
11      db1.style.color = "aqua";
12    }
13  }
14 });
```

- db1-től db9-ig vannak a dobozk az amőbában, mindegyik doboz kattintásakor végbemennek ezek a cselekedetek.

- Nem egy folyamatosan futó ciklus van, hanem mindig kattintáskor történik meg a megvizsgálás, ezáltal az eszközünk nem lesz annyira túlterhelve.



```
1   if (jatekVege === true) {
2     db1.style.cursor = "default";
3     db2.style.cursor = "default";
4     db3.style.cursor = "default";
5     db4.style.cursor = "default";
6     db5.style.cursor = "default";
7     db6.style.cursor = "default";
8     db7.style.cursor = "default";
9     db8.style.cursor = "default";
10    db9.style.cursor = "default";
11    UJRAKEZDES.style.visibility = "visible";
12  }
13  checkPontok();
14 }
```

- Ha vége van a játéknak, akkor a benne lévő utasítások történnek.



```
1  UJRAKEZDES.addEventListener("click", function () {
2      jatekVege = false;
3      db1.textContent = "";
4      db2.textContent = "";
5      db3.textContent = "";
6      db4.textContent = "";
7      db5.textContent = "";
8      db6.textContent = "";
9      db7.textContent = "";
10     db8.textContent = "";
11     db9.textContent = "";
12     db1.style.backgroundColor = "#485460";
13     db2.style.backgroundColor = "#485460";
14     db3.style.backgroundColor = "#485460";
15     db4.style.backgroundColor = "#485460";
16     db5.style.backgroundColor = "#485460";
17     db6.style.backgroundColor = "#485460";
18     db7.style.backgroundColor = "#485460";
19     db8.style.backgroundColor = "#485460";
20     db9.style.backgroundColor = "#485460";
21     db1.style.cursor = "pointer";
22     db2.style.cursor = "pointer";
23     db3.style.cursor = "pointer";
24     db4.style.cursor = "pointer";
25     db5.style.cursor = "pointer";
26     db6.style.cursor = "pointer";
27     db7.style.cursor = "pointer";
28     db8.style.cursor = "pointer";
29     db9.style.cursor = "pointer";
30     GYOZTES.textContent = "";
31     UJRAKEZDES.style.visibility = "hidden";
32 });
33 }
```

- Ha valaki úgy döntene, hogy újramezdi a játékot, minden visszaáll az alap állapotba, a pontok megmaradnak.



Mi az a HTML?

A **HTML (HyperText Markup Language)** egy jelölőnyelv, amelyet weboldalak **szerkezetének felépítésére** használnak.

Nem programozási nyelv, hanem egy olyan leíró nyelv, amely megmondja a böngészőnek, hogy milyen elemek jelenjenek meg az oldalon (szöveg, képek, gombok, linkek stb.).



Mikor jelent meg?

1991-ben Tim Berners-Lee hozta létre, azóta több verziója jelent meg, legújabb szabványa a HTML5.

Hol használják?

Webfejlesztés, minden weboldal alapja.

Mit csinál?

Meghatározza az oldal szerkezetét, szövegeket, címsorokat, képeket jelenít meg, linkeket hoz létre más oldalakra, űrlapokat és gombokat kezel.

Összefoglalva:

A web alapja, nélküle nem léteznek a weboldalak.



Mi az a CSS?

A **CSS (Cascading Style Sheets)** egy stílusleíró nyelv, amely a weboldalak **kinézetéért és megjelenéséért** felel.

A HTML adja a szerkezetet, a CSS pedig megmondja, **hogyan nézzen ki** az oldal (színek, betűk, elrendezés, animációk).



Mikor jelent meg?

Első verziója 1996-ban jelent meg, a World Wide Web Consortium szabványosította.

Hol használják?

Webfejlesztés, weboldalak modernizálása.

Mit csinál?

Színezi és formázza a szövegeket, elrendezi az elemeket, reszponzív dizájnt készít, animációkat és átmeneteket hoz létre.

Összefoglalva:

Széppé és átláthatóvá teszi a weboldalakat.



Mi az a JavaScript?

A **JavaScript** egy programozási nyelv, amely a weboldalak **működéséért és interaktivitásáért** felel.

Míg a HTML a szerkezetet, a CSS a kinézetet adja, a JavaScript teszi **élettel telivé** az oldalt (gombok, űrlapok, animációk, adatkezelés).



Mikor jelent meg?

Brendan Eich hozta létre 1995-ben mindössze 10 nap alatt. Később az ECMA International szabványosította.

Hol használják?

Webfejlesztés, weboldalak funkcionalitására.

Mit csinál?

Reagál felhasználói műveletekre (kattintás, gépelés), dinamikusan módosítja a weboldalt, adatokat tölt le és küld a szervernek, animációkat és effekteket kezel.

Összefoglalva:

A web motorja, funkcionalitást biztosít az weboldalnak.



Konklúzió és önreflexió

Ssz.	Tevékenységek	Igen	Nem	Szöveges kieg.
1.	A projekt modul dokumentációját az elvégzendő feladatokat értettem	X		
2.	A feladat végrehajtásához minden eszközt ismertem	X		
3.	A programozáshoz kapcsolódó biztonsági előírásokat ismerem, és végrehajtottam	X		
4.	Nem okozott gondot a programkód megírása	X		
5.	A projekt programfuttatási része jól működött	X		

- Az amőba játék elkészítése során jobban beleláttam a JavaScript működésébe és annak használatába. Megtapasztaltam, hogyan lehet felépíteni applikációkat webes felületen.

