 Eötvös Lóránd Tudományegyetem

Informatikai Kar

Média- és Oktatásinformatika Tanszék

**Online valós idejű torpedójáték**

**Heizlerné Bakonyi Viktória Major Gábor**

Mester Tanár Programtervező informatikus Bsc

Budapest, 2018

Tartalom

[Bevezetés 4](#_Toc513582596)

[A feladat leírása 4](#_Toc513582597)

[A témaválasztás indoklása 4](#_Toc513582598)

[Felhasználói dokumentáció 5](#_Toc513582599)

[Célkitűzés 5](#_Toc513582600)

[A program telepítése 5](#_Toc513582601)

[Rendszer követelmények a host-hoz 5](#_Toc513582602)

[Backend 6](#_Toc513582603)

[Frontend és Websocket 9](#_Toc513582604)

[IP-cím konfigurálása 10](#_Toc513582605)

[Rendszer követelmények a felhasználók gépéhez 11](#_Toc513582606)

[A program használata 11](#_Toc513582607)

[Regisztráció 12](#_Toc513582608)

[Bejelentkezés 14](#_Toc513582609)

[Kijelentkezés 14](#_Toc513582610)

[Scoreboard és statisztika 14](#_Toc513582611)

[Profil 15](#_Toc513582612)

[Játék indítása 15](#_Toc513582613)

[Játék menete 17](#_Toc513582614)

[Fejlesztői dokumentáció 21](#_Toc513582615)

[Specifikáció 21](#_Toc513582616)

[Fejlesztői környezet 21](#_Toc513582617)

[Feladat elemzése 21](#_Toc513582618)

[Alkalmazott technológiák 21](#_Toc513582619)

[Backend 21](#_Toc513582620)

[Az adatbázis 21](#_Toc513582621)

[Végpontok 21](#_Toc513582622)

[Authentikáció 21](#_Toc513582623)

[Frontend 21](#_Toc513582624)

[Felhasznált könyvtárak 21](#_Toc513582625)

[Segéd függvények 21](#_Toc513582626)

[Komponensek 21](#_Toc513582627)

[Routing 21](#_Toc513582628)

[Stílus szabályok generálása 21](#_Toc513582629)

[Websocket 21](#_Toc513582630)

[Tesztelés 22](#_Toc513582631)

[Backend 22](#_Toc513582632)

[Frontend 22](#_Toc513582633)

# Bevezetés

## A feladat leírása

Az általam megvalósított program egy online valós idejű torpedó játék, ahol a játékosok más, a hálózaton (akár Internet) lévő játékosok ellen tudnak játszani. A játékhoz regisztráció szükséges, ahol a felhasználó a személyes adatai megadása mellet megadja a fiókhoz tartozó felhasználói nevet, ami egyedi és a többiek számára látható. Bejelentkezés után jelentkezhet játékra, amikor is a szerver megpróbál ellenfelet találni számára, ehhez legalább két játékra jelentkezett felhasználó kell. A program nyomon követi a játékra jelentkezettek számát és ezt az információt mindig megosztja a már jelentkezett játékosokkal, így a felhasználó láthatja, hogy vár-e még valaki rajta kívül a játékra.

A játék a szabályoknak megfelelően zajlik: a játékosok elhelyezik a hajóikat az erre kijelölt játéktéren és utána megpróbálják elsüllyeszteni egymás hajóit. A játék folyamatosságát időzítők biztosítják, így elkerülhető a fölösleges várakozás az ellenfélre. A csata addig folyik, amíg az egyik játékos el nem süllyeszti az ellenfél hajóit, vagy valaki fel nem adja.

A csata után a felhasználóhoz tartozó statisztikák, a végeredmény függvényében frissülnek. Ezeket bárki akár bejelentkezés nélkül is megtekintheti az oldalon. Azért, hogy fölöslegesen hosszú listákat elkerüljük minden kategóriából (legtöbb: csata, győzelem, pont) csak az első húsz játékost jelenik meg.

## A témaválasztás indoklása

A webes világ engem mindig is lenyűgözött, különösen az tetszik benne, hogy a kódot egy böngészőben futtatva szemmel láthatóvá válik a munkánk. A Webfejlesztés I-II tantárgyakat teljesítve, csak még inkább kedvem lett az informatika ezen területével foglalkozni, kiváltképpen a frontend-el.

A téma választásakor számomra elsődleges szempont volt egy olyan program készítése, amelyet mások is szívesen használnának, ezért döntöttem a játékprogram készítése mellet.

# Felhasználói dokumentáció

## Célkitűzés

A program célja, hogy megvalósítsa a népszerű torpedó játékot egy online felületen két játékos között, valós időben, mindezt úgy, hogy a lehető legtöbb felhasználót tudja bevonni. Ennek értelmében a játékosok számára minél egyszerűbb és könnyen kezelhető felületet kell biztosítani a lehető legjobb játékélmény eléréséhez.

Továbbá biztosítania kell, hogy a felhasználók tudjanak regisztrálni, be-, illetve kijelentkezni az oldalról. E mellet szeretnénk statisztikát vezetni a felhasználókról. Az ehhez szükséges információkat és szolgáltatásokat, a host gépen futó programokkal kívánjuk ellátni, hogy a felhasználóknak ne kellje további programokat telepíteniük.

## A program telepítése

### Rendszer követelmények a host-hoz

Az alkalmazás nem operációs rendszer specifikus. A futtatáshoz szükséges programok a PHP v.7.2.2, MySQL v.10.1.30-MariDB, Node.js v.8.11.1 és npm v.5.6.0. Nem kizárt, hogy korábbi verziókkal is működne a program, de az elérhető legfrissebb szoftverek használata ajánlott.

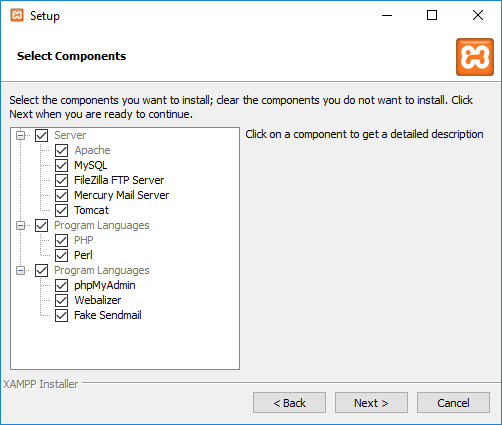
A program működéséhez aktív internet kapcsolat szükséges. Telepítéskor a node-modulok letöltéséhez, azt követőleg pedig a széleskörű elérhetőséghez javasolt. A minél felhasználó barátibb elérhetéshez ajánlott a domain cím használata (pl: www.torpedojatek.hu).

A program nem igényel különleges hardvereket, mint például dedikált videókártya vagy hangkártya.

### Backend

A backend működéséhez szükség lesz a PHP és MySQL programokra. Ezen programokat legegyszerűbben a XAMPP Apache ingyenes disztribúció letöltésével biztosíthatjuk (https://www.apachefriends.org/hu/index.html).

Töltsük le az oldalról az operációs rendszerünknek megfelelő installáló programot, majd kövessük a program utasításait. Vigyázzunk, hogy a program telepítése során a megfelelő komponensek is letöltésre kerüljenek. Ezt a program alapból beállítja, de mindenképpen ellenőrizzük.

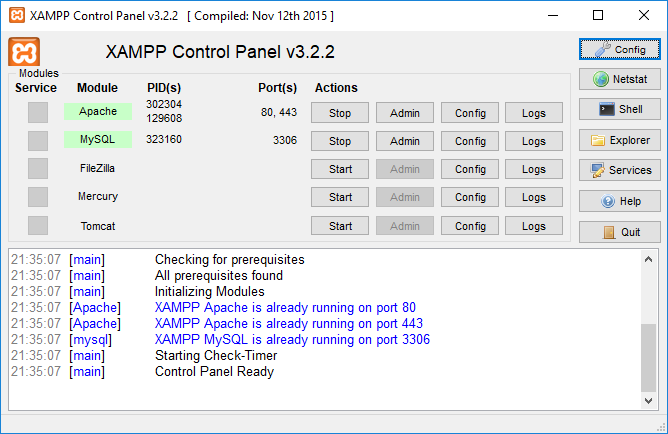


Windows esetén ha mást nem állítottunk be akkor a program alapértelmezetten a **C:\xammpp** mappába települ. Ezen mappán belül találunk egy **htdocs** nevű mappát. Az ebben a mappában lévő fájlokat és könyvtárakat az Apache szerver indítása után a **localhost** **80**-as port-ján érjük el. Ebbe a mappába kell elhelyeznünk a forrás fájlokat.

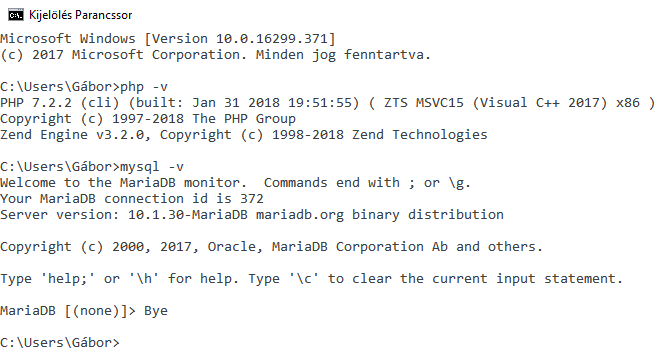
Az adatbázist elkészítő PHP **fájlokat** parancssorból kell futtatni, ehhez szükséges a PHP **path** beállítása.

Windows 10 esetén nyomjuk meg a windows billentyűt ( enter image description here ) és kezdjük el gépelni a „környezeti változók” szót. A megjelenő listáról válaszuk ki az „A rendszer környezeti változóinak módosítása” opciót. Ezt követően a felugró ablakban válasszuk ki a „Speciális” fület, majd kattintsunk a „Környezeti változók…” gombra. Itt válaszuk ki a „Path” nevű változót és nyomjunk a „Szerkesztés…” gombra. Adjuk hozzá a Path-hoz a **C:\xamppp\php** útvonalat, majd minden megnyitott ablakot az „OK” gomb segítségével zárjunk be. Ha volt nyitva parancssorunk vagy PowerShell ablakunk, akkor zárjuk be, és indítsuk újra őket.

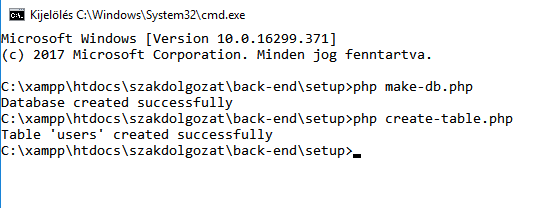
A további műveletekhez el kell indítanunk az Apache és MySQL szolgáltatásokat. Ehhez csak nyissuk meg a XAMPP-ot (**xampp.exe**) és a „Start” gomb megnyomásával indítsuk el az előbbieket. Figyeljük meg, hogy az Apache a 80-as, míg a MySQL a 3306-os port-on működik. Vigyázzunk arra, hogy ne akadjanak össze más szolgáltatásokkal a gépünkön.



Az eddigi lépések sikerességéről, úgy győződhetünk meg, ha parancssort vagy PowerShell ablakot nyitunk és begépeljük a **php -v** illetve **mysql -v** parancsokat. Ezek a parancsok megmondják, hogy az adott programból hányas verzióval rendelkezünk. Ha valami folytán nem sikerült volna megfelelően telepítenünk vagy beállítanunk valamelyik programot, akkor hibaüzenetet kapunk. Az utolsó parancs után egy ctrl + C billentyű kombináció beütése szükséges. Ha minden sikerült az alábbit kell látnunk:



A telepítés során következő lépés az adatbázis létrehozása és beállítása. A **htdocs** mappán belül forrás fájlok között lépjünk be a **back-end** nevű mappába. Itt található a PHP szerver forráskódja. A **back-end\setup** mappában nyissunk meg a parancssort és futtassuk a **make-db.php** fájlt (**php make-db.php**), ez elkészíti számunkra a működéshez szükséges adatbázist. Ezt követően járjunk el hasonlóan a **create-table.php** fájllal, ami az adatbázisban elkészíti a „**user**” nevű táblát, amiben a felhasználóink adatait tároljuk. Siker esetén a következőt kell látnunk:

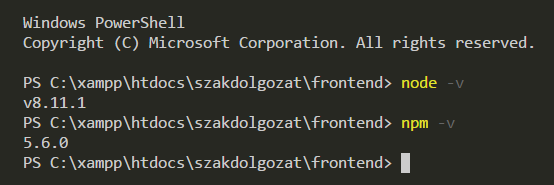


Az adatbázis sikeres létrehozásáról úgy győződhetőnk meg, hogy egy böngészőt nyitunk, majd az url mezőbe a **localhost:80/phpmyadmin/** címet írjuk. Itt a phpMyAdmin felületén nyomon követhetjük és szerkeszthetjük az adatbázisunkat, bár utóbbi művelet NEM ajánlott.

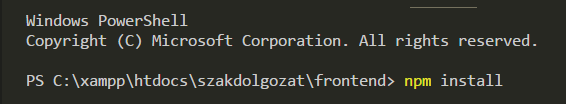
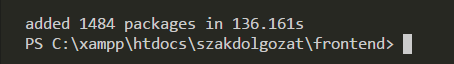
### Frontend és Websocket

A frontend telepítéséhez elsőként le kell töltenünk a Node.js, ezt a programot a https://nodejs.org oldalról tudjuk ingyenesen beszerezni. Válasszuk a „Recommended For Most Users” opciót és töltsük le.

A telepítés során kövessük az installáló szoftver utasításait és ne változtassuk meg az alapértelmezett beállításokat. A telepítés sikerességéről, úgy győződhetünk meg, ha nyitunk egy parancssort vagy PowerShell ablakot és begépeljük a **node -v** és **npm -v** parancsokat. Ha minden rendben van, akkor az alábbit kell látnunk:



Siker esetén installálhatjuk az npm csomagokat. A **frontend** mappába nyissunk PowerShell ablakot és futtassuk az **npm install** parancsot. Ez telepíti a **node\_modules** mappát, amiben a frontend és a websocket működéséhez szükséges könyvtárak találhatóak.

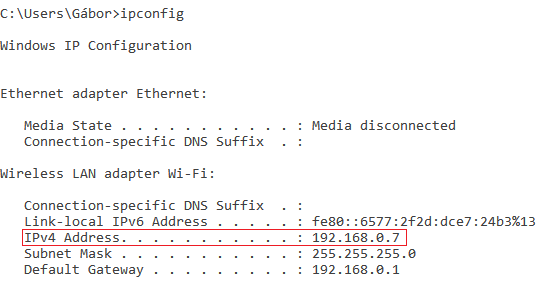
Ha nincs hibaüzenet, akkor az installáció sikeresen befejeződött. A frontend elindításához PowerShell ablakban adjuk ki az **npm start** parancsot, ekkor az alapértelmezett böngésző megnyílik a **localhost** 3000-es port-on, ahol elérhetővé válik a játék weboldala.

A frontend-hez készültek automatizált tesztek is. Ezek a fájlok érdemben nem tesznek hozzá a program működéséhez, cserébe azonban biztosítják az oldal megfelelő működését, illetve segítik a kód működésének a megértését. További előnyük, hogy későbbi továbbfejlesztés vagy módosítás esetén megvédik a hibáktól a már meglévő kódbázist. A teszteket a PowerShell ablakban az **npm test** parancs kiadásával lehet elindítani.

### IP-cím konfigurálása

A forráskód úgy lett megírva, hogy a szervert és a websocket-et **localhost** (azaz **127.0.0.1**) címen keresi, így a program csak az adott számítógépen működik, kívülről más gépek által nem elérhető. Ez azért van így, mert a célgép IP-címe nem ismert a program írásakor. Hogy ezt a problémát megoldjuk és mások által is elérhetővé tegyük a programot be kell állítanunk a host gép IP-címét.

Ha nem tudjuk fejből a saját IP-címünket akkor nyissunk egy parancs sort és futtassuk az **ipconfig** parancsot. Itt keressük meg az eszközünk IPv4 címét:



Ezt a számot kell (esetünkben: **192.168.0.7**) beírni a **localhost** helyett a backend-ben a **back-end\private\headers.php** harmadik sorban és frontend-ben két helyen a **frontend\src\actions\fetchAjax.js** huszadik sorában és a **frontend\src\actions\wsclient.js** harmadik sorában vigyázva arra, hogy ezen kívül semmit nem változtassunk.


Ezzel biztosítjuk, hogy mások által is elérhető legyen a játék. Érdemes ekkor már a test fájlok között is ezt az átírást megtenni, mivel e-nélkül a **fetchAjax**-ra vonatkozó tesztek elbuknak. Ezeket a teszteket a **frontend\src\tests\fetchAjax.test.js** fájlban találjuk.

Az IP konfiguráció véglegesítéséhez újra kell indítani a frontend-et (crlr+C, majd **npm start**).

### Rendszer követelmények a felhasználók gépéhez

A felhasználó részéről nincs szükség program telepítésére. Egy internet böngésző elegendő. Javasolt program: Chrome v.68 vagy egy újabb verzió. Természetesen feltételezzük, hogy a felhasználó eszköze azonos hálózaton van a host eszközzel.

## A program használata

Ha a működéséhez szükséges szolgáltatásokat elindítottuk a host gépen, akkor a **localhost** vagy az általunk beállított IP-cím 3000-es port-ján érjük el a programot (**localhost:3000/** vagy esetünkben **192.168.0.7:3000/**). Ezt a böngésző url mezőjébe kell írni.

A mai trendeknek megfelelően a weboldal reszponzív, ami azt jelenti, hogy az oldal alkalmazkodik az eszköz képernyő méretéhez. Erre azért van szükség mivel az oldal ugyanúgy látogatható okostelefonról, vagy tabletpcről, mint asztali számítógépről.

### Regisztráció

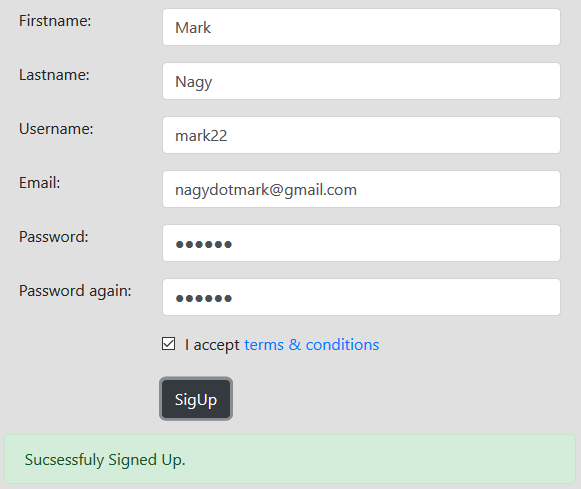
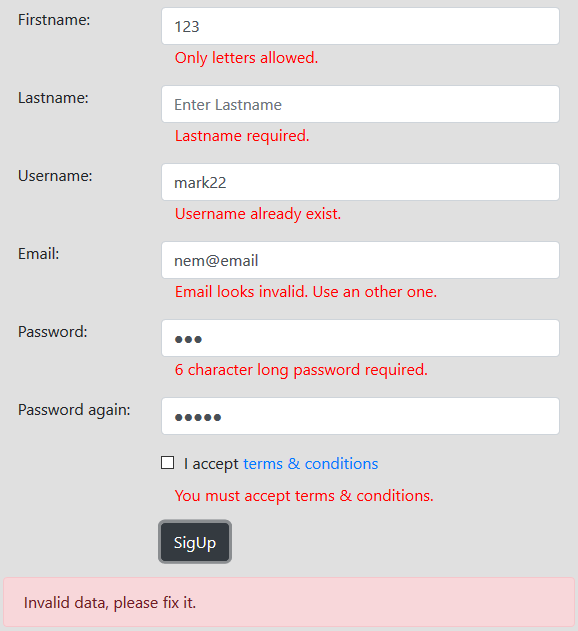
Regisztrálni a **/signup** címen lehet, ez csak akkor lehetséges, ha az oldalon nem vagyun bejelentkezve egy másik felhasználóval. A regisztrációhoz szükséges megadnunk a kereszt- és vezetéknevünket, egy egyedi felhasználónevet, ami a továbbiakban már nem módosítható, egy email címet, illetve jelszót kell választanunk.



A beadott adatok ellenőrzését a szerver végzi. Hibás adat esetén a felhasználó hibaüzenetet kap, ami jó látható módon, a hibás beviteli mező alatt pirossal jelenik meg. Ilyen hiba lehet:

* ha egy mezőt üresen hagyunk,
* ha a kereszt- és/vagy a vezetéknévben számot, szóközt vagy speciális karaktereket használunk (ide értve az ékezetes betűket is),
* ha nem egyedi vagy legalább hat karakter hosszú felhasználónevet választunk,
* ha az e-mail cím nem megfelelő formátumú,
* ha az általunk megadott jelszó nem legalább hat karakter hosszúságú vagy nem egyezik meg a második megadással
* ha nem fogadjuk el a végfelhasználói feltételeket

Sikeres regisztráció esetén piros hibaüzenet helyett egy zöld mező jelenik meg az űrlap alján, amely közli velünk a regisztráció sikerét. Ekkor egy kis idő elteltével a program átirányít minket a bejelentkezési oldalra.

### Bejelentkezés

A **/login** címen található bejelentkezés a regisztrációhoz hasonló űrlapon történik. Itt a felhasználónevünket és a jelszavunkat kell megadunk. Itt hibaüzenetet akkor kapunk, ha ezt a két mezőt üresen hagyjuk, ha nem létező felhasználónevet, vagy ha rossz jelszót adtunk meg.

Lehetőségünk van arra, hogy az oldal emlékezzen ránk, így a böngésző bezárása és újra nyitása esetén nem kell újra bejelentkeznünk, (biztonsági okokból 30 naponta meg kell ismételni a bejelentkezést).

Sikeres bejelentkezés esetén az űrlap alján erről értesít minket az oldal, majd kis idő elteltével átirányít minket a profilunkhoz.

### Kijelentkezés

Kijelentkezni a menüsor jobb szélén lévő „Logout” link megnyomásával lehet. Sikeres kijelentkezés esetén az oldal átirányít minket a bejelentkező oldalra, ahol üzenetet kapunk a sikeres kilépésről. Ha bejelentkezéskor bejelöltük, hogy az oldal emlékezzen ránk, akkor ezzel a művelettel ezt visszavonjuk, és következő alkalommal be kell jelentkeznünk a felhasználónevünk és jelszavunk megadásával.



### Scoreboard és statisztika

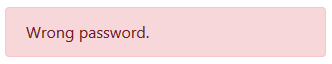
A játék során feljegyzéseket készítünk arról, hogy egy játékos hány játékot játszott, ebből mennyit nyert meg és mennyi pontot gyűjtött össze. Ezeket a felhasználók nyomon követhetik a profiljukban.

Az oldalon elérhető egy lista, a **/scoreboard** címen, amely a kategória (legtöbb csata/győzelem/pont) húsz legjobb játékosának a felhasználó nevét és statisztikáját feltünteti. Az adott kategória megtekintéséhez a táblázat fölötti menüből kell kiválasztani a megfelelő menüpontot. Ez a lista bárki számára megtekinthető, bejelentéseshez nem kötött.

### Profil

A profilunkba belépve lehetőségünk van a regisztrációkor megadott adataink megtekintésére és módosítására (a felhasználónév nem módosítható). Ez a **/account** címen található, és ha bejelentkezés nélkül keressük fel, akkor az oldal átirányít minket a bejelentkezésre.

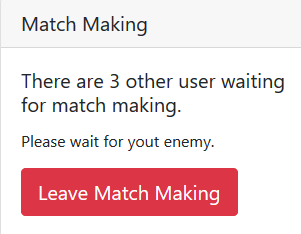
Az adatink a „Profile”, a felhasználónkhoz tartozó statisztika a „Statistics” menüpont alatt tekinthetők meg. Az adatok módosítását a „Modify” menüpont alatt érjük el. Hasonlóan működik az eddigi űrlapokhoz. Itt csak azokat a mezőket kell kitöltenünk, amelyiket módosítani szeretnénk. Az adatok módosítását az aktuális jelszavunkkal meg kell erősítenünk. Jelszómódosítást a „Changepassword” menüpont alatt intézhetjük. Itt biztonsági okokból meg kell adnunk a jelenleg használt jelszavunkat, majd az új jelszót kétszer. Az űrlapok hiba esetén hibaüzenetet küldenek, sikeres módosítást az űrlap alatt zölden megjelenő üzenet jelzi.



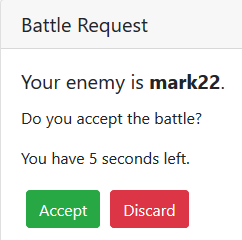
### Játék indítása

Játékot bejelentkezve, a profilunkból tudjuk indítani a „Start Game” gomb megnyomásával. Ekkor bekerülünk a „match making” folyamatba, melynek lényege, hogy a szerver keres számunkra egy másik játékost, aki szintén jelentkezett játékra, magyarán ellenfelet keres nekünk.

Miközben várunk az ellenfelünkre nyomon követhetjük, hogy rajtunk kívül még mennyi játékos vár a sorsolásra. Ez az információ folyamatosan frissül, ha valaki csatlakozik vagy távozik a „match making” folyamatból. Távozni a piros „Leave Match Making” gomb megnyomásával lehet.



Legalább két „match making”-re jelentkezett játékos esetén a program párba tudja állítani őket. Ekkor a játékosok értesítést kapnak arról, hogy kit sorsoltak ki számára ellenfelül. A játék akkor kezdődik, ha a játékosok kölcsönösen elfogadják egymást, mint ellenfél. Elfogadni a zöld „Accept” gomb megnyomásával lehet, erre tíz másodperce van a játékosoknak. Természetesen lehetőség van az ellenfél elutasítására a piros „Discard” gomb megnyomásával. Ekkor mi visszakerülünk a „match making” állapotba, hogy a szerver új ellenfelet sorsolhasson számunkra. Az a játékos akit nem fogadtunk el, mint ellenfél értesítést kap erről, majd ő is visszakerül a „match making” állapotba.



Az időlimit jól látható helyen a gombok felett helyezkedik el. Ha az idő letelt, de a játékos nem nyomott gombot, akkor az automatikusan az ellenfél elutasítását jelenti („Discard”).

Játékra csak olyan játékos jelentkezhet, aki se „match making” állapotban, se játékban nincsen. Ellenkező esetben a gomb megnyomásakor üzenetet kapunk arról, hogy nem kerültünk „match making” állapotba, mivel az előző feltételeknek egyikének nem felelünk meg.

### Játék menete

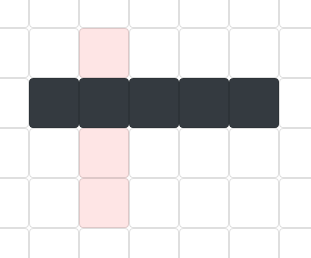
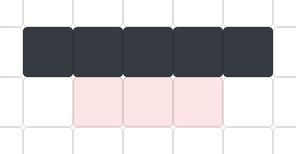
A játék az ellenfelek kölcsönös elfogadása után indul, a **/game** oldalon. Erre az oldalra csak az elfogadást követő lépésben kerülhetünk. A játéktér két nagyobb egységből és a felettük lévő információs sávból áll. Ezen az információs sávon láthatjuk ellenfelünk felhasználónevét, illetve itt található egy „Exit Game” gomb, melynek megnyomásával bármikor véget vethetünk a játéknak, magyarán feladhatjuk azt és visszatérhetünk a profilunkhoz.

Az információs sáv alatt találjuk a játékmezőt. Itt a „Your Ships” nevű tízszer tízes mezőn kell elhelyeznünk a hajóinkat. Egy öt, egy négy, egy három, két kettő és két egy egység hosszú hajót, ebben a sorrendben. A szabályoknak megfelelően hajóinkat függőlegesen és vízszintesen is elhelyezhetjük, viszont nagyon fontos, hogy a hajók egymást nem fedhetik, a játék mezőről ki nem lóghatnak és az oldalaikon nem érhetnek össze.

A program a hajók elhelyezését úgy segíti, hogy halványan megjelöli azokat a mezőket, amelyre a hajó kerülni fog. A hajót bal egér kattintással tudjuk elhelyezni. Ha az irányt szeretnénk módosítani, azt a **szóköz** billentyű lenyomásával tehetjük (miközben a kurzort a mező fölött tartjuk).

Ha egy hajót nem tudunk lehelyezni, mert azzal megsértenénk a játékszabályokat, akkor ezt a program halvány pirossal jelzi. Ekkor hiába kattintunk nem kerül lehelyezésre hajó.

A játékmező alatt három gomb található. Sorra a „Reset Table”, a „Ready” és a „Random” gombok. Az első gomb addig nem kattintható, amíg nincs elhelyezve legalább egy hajó a mezőn. Ezen gomb megnyomásával a játékmezőről eltűnnek az eddig elhelyezett hajók, így újra tudjuk kezdeni a lerakást. Fontos megjegyezni, hogy a korrekció elvégzése a rendelkezésre álló időt nem állítja meg, vagy tekeri vissza. A második gomb akkor válik kattinthatóvá, ha az összes hajót a szabályoknak megfelelően elhelyeztük. Ennek a gombnak a megnyomásával véglegesíthetjük a hajóink pozícióját, ekkor ezek a gombok eltűnnek és le áll a visszaszámláló is. A „Random” gomb megnyomásával a program véletlenszerűen helyezi el számunkra a hajókat.

A hajók elhelyezésére negyvenöt másodperc áll rendelkezésünkre. Ha végeztünk a hajók letételével nyomjuk meg a „Ready” gombot. Ha ezt elmulasztjuk, akkor az idő letelte után a játéknak vége szakad, mint ha megnyomtuk volna az „Exit Game” gombot.

Amikor mindkét játékos jelezte, hogy készen van a hajók elhelyezésével és rákattintott a „Ready”-re, akkor a program véletlenszerűen kisorsolja azt a játékost, aki először kezdheti a torpedók lövését. Ekkor már mindkét játékos számára megjelenik a „Enemy Area” nevű játéktér, majd az információs sávban a játékosok láthatják, az „Enemy turn. Please wait” vagy a „You turn. Fire a missile on the Enemy Area” üzenetet, mely egyértelműen jelzi, hogy melyik játékos következik.

A soron következő játékos az „Enemy Area” nevű játéktéren kiválaszthatja, hogy hova szeretne torpedót indítani. Egy körben csak egy mezőt választhat. Kiválasztani bal egér gomb kattintással lehet, ekkor a mező körüli piros szegély jelenik meg, ami egyértelműen jelzi a választásunkat. A program itt is segít azzal, hogy azt a mezőt, amelyre a kurzorunk mutat, halványan megszínezi.



A választott célpontot mindaddig lehetőségünk van módosítani, amíg meg nem nyomtuk a játéktér alatti található „Fire” gombot. A gomb csak akkor kattintható, ha van kijelölve mező, és csak annak a játékosnak jelenik meg aki éppen következik. A játék folyamatosságát biztosítandó tizenöt másodpercünk van a torpedó célpontját kiválasztani és megnyomni a „Fire” gombot. Ellenkező esetben a játék végeszakad.

A „Fire” gombra kattintva a körünknek vége, ekkor az ellenfelünk következik. Arról, hogy a torpedó eltalálta-e az ellenfél hajóját azonnal üzenetet kapunk. Az üzenet az „Enemy Area” alatt ott jelenik meg, ahol eddig az időzítőt láttuk. Természetesen az ellenfél is értesül a lövés kimenetéről. Ő a „Your Ships” alatt található mezőben olvashatja ezt. A következő üzenetek jelenhetnek meg:

|  |  |
| --- | --- |
| „You MISSED the enemy ship.” | „The enemy MISSED your ship.” |
| „You HIT the enemy ship.” | „The enemy HIT your ship.” |
| „You SANK the enemy ship.” | „The enemy SANK your ship.” |

Az üzeneteken kívül a lövés eredménye természetesen a játék mezőn is megjelölésre kerül, attól függő színnel, hogy a torpedó talált-e vagy sem. Ha talált, akkor piros, ha nem akkor azt kék színnel jelzi a program. A játékosok saját hajóit érő torpedók a „Your Ships”, mezőkön, az általuk kilőtt torpedók pedig az „Enemy Area” mezőkön jelennek meg.

Az alábbi ábrán a játék egy előrehaladott állapota látható. Ha megfigyeljük az **A játékos** által kilőtt torpedókat az (lásd **A játékos** „Enemy Area”) és a B játékos hajóit tartalmazó játékteret (lásd **B játékos** „Your Ships”), akkor jól láthatjuk, hogy a program milyen módon jelzi a torpedó lövések eredményét.

|  |  |
| --- | --- |
| A játékos | B játékos |
|  |  |
|  |  |

A játék végeztével a győztest harminc, ellenfelét tíz ponttal jutalmazza a program. Ezután a profil oldalunkról akár rögtön jelentkezhetünk egy újbóli játékra.

# Fejlesztői dokumentáció

## Specifikáció

### Fejlesztői környezet

### Feladat elemzése

## Alkalmazott technológiák

## Backend

### Az adatbázis

### Végpontok

## Authentikáció

## Frontend

### Felhasznált könyvtárak

### Segéd függvények

### Komponensek

### Routing

### Stílus szabályok generálása

## Websocket

## Tesztelés

### Backend

**API/** végpont:

|  |  |  |
| --- | --- | --- |
| Teszt eset | Elvárt viselkedés | Teszt kimenete |
| A végpontra **GET** metódusú kérést indítunk. Nincs bejelentkezett felhasználó. | A végpont válaszoljon **200**-as response kóddal és a response **user** mezője legyen **null**. | 200 OK,  **user:** null |
| A végpontra **GET** metódusú kérést indítunk. Bejelentkezett felhasználó: mark22. | A végpont válaszoljon **200**-as response kóddal és a response data **user** mezője legyen **mark22**. | 200 OK,  **user:** mark22 |
| A végpontra **POST** metódusú kérést indítunk. | Mivel az adott metódus nem támogatott a végpont válaszoljon **405**-as response kóddal és ne küldjön response data-t. | 405 Method Not Allowed |

**API/auth/login/** végpont:

|  |  |  |
| --- | --- | --- |
| Teszt eset | Elvárt viselkedés | Teszt kimenete |
| A végpontra **GET** metódusú kérést indítunk. Nincs bejelentkezett felhasználó. | A végpont válaszoljon **200**-as response kóddal és a response **message** mezője legyen **null**. | 200 OK,  **message:** null |
| A végpontra **GET** metódusú kérést indítunk. Bejelentkezett felhasználó: mark22. | A végpont válaszoljon **403**-as response kóddal, kóddal és ne küldjön response data-t. | 403 Forbidden |
| A végpontra **POST** metódusú kérést indítunk. **body:** {  **username: ,**  **password:**  }  Nincs bejelentkezett felhasználó. | A végpont válaszoljon **400**-as response kóddal.  Sikertelen bejelentkezés, hibaüzenetek várunk a **username** és a **password** mezőkre. | 400 Bad Request  **success:** false,  **errors:** { **username:** Username required.  **password:** Password required.},  **message:** Failed to LogIn. |
| A végpontra **POST** metódusú kérést indítunk. **body**: {  **username:** mark22,  **password:** notgood  }  Nincs bejelentkezett felhasználó. | A végpont válaszoljon **400**-as response kóddal.  Sikertelen bejelentkezés. | 400 Bad Request  **success:** false,  **message:** Wrong username or password. |
| A végpontra **POST** metódusú kérést indítunk. **body**: {  **username:** mark22,  **password:** 123456  }  Nincs bejelentkezett felhasználó. | A végpont válaszoljon **200**-as response kóddal.  Sikeres bejelentkezés. | 200 OK  **success:** true,  **message:** Sucsessfully LoggedIn. |
| A végpontra **PUT** metódusú kérést indítunk. | Mivel az adott metódus nem támogatott a végpont válaszoljon **405**-as response kóddal és ne küldjön response data-t. | 405 Method Not Allowed |

### Frontend

>> auto tesztek