fejlesztőibe

Egy ilyen játék webes elkészítése során természetesen egy full-stack alkalmazás lesz a végtermék, amely az internetes technológiák spektrumának szinte egészét lefedi. Kliensoldalon szükséges megtervezni minden arculatot, a játék kinézetét. Ehhez a JavaScript programozási nyelv egyik nyílt forráskódú könyvtárát vettem segítségül, a ReactJs keretrendszert. Nagyon kényelmes eszközöket ad játékok implementálásához. Egyetemi éveim során kezdtem el használni szintén egy társasjáték elkészítésének megvalósításához. Az elkészítéshez szükséges technológiák között feltétlenül helyet foglal egy szerveroldali megoldás is, ami kapcsolatot tart a játékban keletkező adat és az adatbázis között, amiben előbbieket tárolom. Ezt NestJS-ben és SQLite-ban valósítottam meg, amikhez MikroORM mappert használtam. A mapper teszi könnyűvé a kommunikációt kód szintjén az adatbázissal. Nem lenne élvezetes a játék egyedül, különben ki piszkálna bele a térképünkbe (erről később szó lesz). Ennek a programozására pedig websocket technológiát vettem igénybe, ez tartja a kapcsolatot valós időben a játékosok között.

Először a backend technológia implementálásával kezdtem, lefektettem a táblákat, és a REST-API hívásokat, a CRUD összes műveletére. Amikor már ott tartottam, hogy Postman-nel – egy API tesztelő programmal – le tudtam szimulálni az összes létező szerverhívást, ami a játékban előfordulhat, elkezdtem dolgozni a kliensoldalon. Természetesen sokszor vissza-vissza kellett térnem a backendhez, hogy igazítsam a frontendhez, nehéz tökéletesen megtervezni előre egy backend kódot. A websocket rész elkészítésére akkor tértem át, amikor a frontendnek már a konkrét játék részénél jártam.

A feladatomat akkor tekintettem megoldottnak, amikor ezek a technológiák körbeértek, tehát a játék a frontend kihasználta a backend összes megírt szükséges funkcióját, beleértve a játék végén kialakult pontszámok, helyezések adatbázisba való tárolását is.

Ha a játékot valaki üzemeltetni szeretné, akkor a forráskód letöltése után (ami megtalálható a következő github linken: <https://github.com/PettkoHarsanyi/kartografusok>) egy olyan mapparendszer birtokában lesz, amiben megtalálható a játék működtetéséhez szükséges három technológia. Mind három technológiát egy úgynevezett „npm csomagkezelő” kezel, aminek megléte szükséges az üzemeltető rendszerén. Ezt legkönnyebben a <https://nodejs.org/en/download/> oldalról lehet beszerezni, ahol a „Node.js”-el együtt települ az „npm” is. A program futtatása előtt még egy dolog szükséges, méghozzá telepíteni a program függőségeit. Ennek egyszerűsítésére elhelyeztem egy „install.bat” nevű programot, ami mindhárom technológia függőségeit telepíti, különben egyenként kéne a három mappában kiadni külön-külön az „npm i” parancsot. A telepítő befejeztével a program egy szintén előre megírt futtató program segítségével tud elindulni. Ez szintén a program gyökérkönyvtárában található „start.bat” program. Ennek hiányában a három technológia könyvtárában külön-külön lefuttatott „npm start” paranccsal hasonló eredményeket érhetünk el.

ADATB elérés

Bizonyos lépések végrehajtása után (amikről a fejlesztői dokumentáció részben szó lesz) a játék a böngésző keresőjébe beírt <http://localhost:3001> címen érhető el a felhasználó számára.