



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

Informatika Kar

Média- és Oktatásinformatika Tanszék

Educative Gamification

Témavezető:

Bende Imre

Tanársegéd, MA.

Szerző:

Juhász Alexandra Nelli

Programtervező informatikus BSc.

Budapest, 2023

Témabejelentő

A témám az „Educative Gamification”, azaz Játszva Tanulás. A céлом egy olyan játék elkészítése, amely során számtalan pénzügyi akadállyal/kérdéssel kell szembe néznünk és döntéseket hoznunk, melyek befolyásolják a játék kimenetelét.

Kezdetben kapunk egy életcél, illetve egy karakter jellemet, azaz kezdő pénzt, munkahelyet, alap tulajdonságokat. Női és férfi karakter közül lehet választani és nevet adni neki. Ő lesz az, akinek segítenünk kell elérni a célját a lehető leghamarabb, figyelembe véve a képességeit, állapotát és élethelyzetét.

Az utunk során különféle kisebb játékok vállnak elérhetővé, melyek teljesítésével emberi képességekre tehetünk szert (pl. szerencse, intelligencia, stb...), amik könnyíthetik a célunkhoz vezető utat. Véletlenszerű események is szerveződhetnek vagy nem várt dolgok történhetnek, melyek nagyban befolyásolhatják az életkörülményünket (pl. szerencsejáték, egészségügyi kiadások, munkahelyi gondok, tárgyak/pénz találása, stb...).

A játék döntési fákon alapul, mely tárolja a kérdéseket/problémákat és a hozzátartozó lehetőségeket, majd a döntés alapján következnek az utána való akadályok, illetve tartalmaz jogosultság szerinti autentikációt (admin, player). Lehetőségünk van további döntési fa létrehozására és annak végig játszására, vagy véletlenszerű események és azok esélyeinek, hatásainak bővítésére is.

A játék React Js és egyéb webes technológiák (HTML, CSS, PHP) segítségével készül, az adatok tárolása MySQL adatbázisban történik.

Tartalomjegyzék

Köszönnetnyilvánítás	5
1. Bevezetés	6
2. Felhasználói dokumentáció.....	7
2.1 Rendszerkövetelmények	7
2.1.1 Frontend.....	7
2.1.2 Backend	8
2.2 A játék használata	9
2.2.1 Felhasználói kezdőlap	9
2.2.2 Bejelentkezés.....	9
2.2.3 Rendszergazda kezdőlap	10
2.2.4 JátéKFelület.....	14
3. Fejlesztői dokumentáció	29
3.1 Fejlesztői környezet	29
3.2 Drótváztervek	29
3.2.1 Kezdőlap.....	29
3.2.2 Adatbázis szerkesztő	31
3.2.3 JátéKFelület.....	32
3.3 Grafika.....	33
3.3.1 Karakterek	33
3.3.2 Hátterek	35
3.3.3 Ikonok és gombok	36
3.4 Használati esetek	38
3.5 Architektúra.....	39
3.6 Adatbázis felépítése	41
3.7 Backend.....	43
3.7.1 Játékos adatainak beszúrása	43

3.7.2 Feladatok szerkesztése.....	43
3.7.3 Munkák szerkesztése	45
3.7.4 Fizetések szerkesztése	45
3.7.5 Egyéb lekérdezések	45
3.8 Frontend	47
3.8.1 Felhasználói kezdőlap	47
3.8.2 Rendszergazdai kezdőlap	48
3.8.3 Belépési felület	48
3.8.4 Útvonalak	48
3.8.5 Komponensek	49
3.8.6 Modell.....	53
3.9 Bővítési lehetőségek.....	63
3.10 Tesztelés	64
3.10.1 Manuális tesztek	64
4. Forrásjegyzet.....	73

Köszönetnyilvánítás

Köszönöm szépen Csorba Gergelynek a rengeteg segítséget, aki inspirálta a szakdolgozatom témáját és részletes bemutatást tartott róla. Hálás vagyok, hogy vállalta a játék design elemeinek elkészítését.

1. Bevezetés

A szakdolgozatomban bemutatott „Invesztopolisz” egy olyan számítógépre tervezett játék, ami a valóságban is előforduló pénzügyi döntések elé állítja a játékost. A döntések befolyásolják a játék kimenetelét, így a karakterünk lehet nagyon gazdag, vagy akár szegény is. A célja az új dolgok élményszerű megismerése, melyekhez pozitív emlékek kötnek. Aktív tanulást biztosít számunkra, hogy jobban berögzüljenek az információk, mint passzív tanulásnál.

Az alábbiakban gyorsan tisztázom a passzív és az aktív tanulást:

- Passzív tanulás: Hagyományos tanulási módszer, ahol a tanulók bevonása az órán nagyon kevés, tehát tanár központú. A tanár leadja az anyagot, míg a tanulók hallgatnak vagy jegyzetelnek. Lehetővé teszi az ismeretek gyors bemutatását, viszont a tanulóknak a leadott és hallott anyagot kell megtanulniuk maguktól, ami többségében felületes és lassú.
- Aktív tanulás: A tanulók be vannak vonva az órába, a tanárral együtt ötletelnek és dolgoznak ki megoldásokat. Észrevétlenül ösztönzik a tanulót a gondolkodásra, illetve növeli a figyelmét, mely hatására sokkal jobban berögzül a leadott anyag. Kevesebb információ kerül bemutatásra, ellenben sokkal hatásosabb a tanuló számára.

Az egyik legelső ilyen oktató játék, a „*The Oregon Trail*”¹ 1971-ben jelent meg, amit iskolákban is engedélyeztek telepíteni. A játék célja, hogy egy csapat telepessel eljuss Oregonba. A telepesek nehéz életét mutatja be a játék, és hogy nem minden a terv szerint alakultak a dolgok, mert folyamatosan fogyott az élelem és a különböző erőforrások, nem várt természeti eseményekkel, támadásokkal és betegségekkel kellett szembe nézniük. Tapasztalom szerint sokkal könnyebben és hatékonyabban lehet tanulni játékokkal, elősegíti az új ismeretek, vagy új nyelvek megismerését.

Az általam szakdolgozatként megvalósított játék a felhasználókat valós pénzügyi döntések elé helyezi, tanulva azok következményeiről, szert téve pénzügyi tudásra.

Lényegében ez egy pénzügyet oktató játék, ahol megismerhetjük az alapfogalmakat és befektetési formákat.

¹ The Oregon Trail: [https://en.wikipedia.org/wiki/The_Oregon_Trail_\(series\)](https://en.wikipedia.org/wiki/The_Oregon_Trail_(series))

2. Felhasználói dokumentáció

2.1 Rendszerkövetelmények

Az „Invesztropolisz” játék tesztelése elsősorban 1920 x 987 felbontású Google Chrome-mal (113.0.5672.63) történt, így ez az elsődlegesen támogatott böngésző. A játék 1920 x 1080 felbontású, Windows 10-es operációs rendszerű számítógépre készült, tehát telefonon vagy kisebb felbontású számítógépen nem alkalmas játszani vele. Ha a felhasználó a forráskód alapján szeretné futtatni a játékot lokális környezetben, akkor a leírtak szerint megteheti. Az alkalmazás kódja GitHub-on elérhető a következő linkeken:

- <https://github.com/juhasnellee/szakdoga-frontend>
- <https://github.com/juhasnellee/szakdoga-backend>

Mindenek előtt fel kell telepítenünk a szükséges alkalmazásokat, melyek a linkre kattintva elérhetőek:

- Node - <https://nodejs.org/en/download/>
- Xampp - <https://www.apachefriends.org/download.html>

2.1.1 Frontend

Nyissuk meg a parancssort, és futtassuk le az alábbi parancsot, hogy majd létre tudjunk hozni egy új React projektet (*ha már fel van telepítve a „create-react-app”, akkor töröljük ezzel a parancssal: npm uninstall -g create-react-app, és telepítsük vissza*):

1. npm install -g create-react-app

Az előző parancssort bezárhatjuk. Hozzunk létre bárhova egy üres mappát, majd felül a mappa útvonalába kattintva írjuk be a következőt: „cmd” és nyomjunk egy ’Enter’-t. Ekkor megnyílik egy új parancssor, ahol futtassuk le az alábbi parancsokat, hogy létrehozzuk a projektünket és letöltsük a szükséges bővítményeket:

2. npx create-react-app my_react_app - „my_react_app” a projekt neve
3. cd my_react_app - „my_react_app” a projekt neve
4. npm install react-router-dom
5. npm install --save react-wheel-of-prizes --force
6. npm install semantic-ui-react semantic-ui-css --force

Töltsük le a *szakdoga-frontend* nevű repositoryt (jobb felső sarok zöld „Code” gombra kattintva), majd kicsomagolás után az *src* mappájával írjuk felül az újonnan létrehozott React projektünk *src* mappáját. Majd indítsuk el az alkalmazásunkat az utolsó parancssal:

7. npm start

Fontos, hogy az utolsó parancs kiadása után NE zárjuk be a terminált!

2.1.2 Backend

Miután feltelepítettük a Xampp-ot, keressük meg a telepítési helyét, majd a mappák között a *htdocs* nevű mappát. Hozzunk létre egy *szakdoga-backend* nevezetű üres mappát. Töltsük le a *szakdoga-backend* nevű repositoryt (jobb felső sarok zöld „Code” gombra kattintva), majd kicsomagolás után másoljuk be az üres mappába a letöltött fileokat.

Ezután nyissuk meg a Xampp Control Panelt, indítsuk el „Apache”-t és a „MySQL”-t a mellettük lévő „Start” gomb megnyomásával. Ez után nyomjuk meg a „MySQL” mellett lévő „Admin” gombot, ami egy „phpMyAdmin” adatbázist tölt be. Itt létre kell hoznunk egy adatbázist „educ_gamific” néven az alap formátummal, amire az alkalmazás jobb oldalán van lehetőségünk (lásd 1. ábra).



1. ábra: Adatbázis létrehozása

Ha ez megvan, lépjünk be a létrehozott adatbázisba és a fejlécen kattintsunk az „Importálás” menüpontra. Az importálandó fájlhoz válasszuk ki a *szakdoga-backend* mappában található *educ-gamific.sql* fájlt, majd nyomjuk meg alul az „Importálás” gombot.

Az utasításokat végrehajtva a <http://localhost:3000/> linken fog futni a játék a továbbiakban.

2.2 A játék használata

2.2.1 Felhasználói kezdőlap

Ahogy megnyitjuk a játékot, a 2. ábrán látható kép fogad minket. A jobb felső sarokban a „Bejelentkezés” gombbal a rendszergazdai kezdőlapot érhetjük el, ahol módosítani tudjuk az adatbázisban tárolt elemeket. Ez csak akkor lehetséges, ha rendelkezünk a megfelelő adatokkal.



2. ábra: Felhasználói kezdőlap

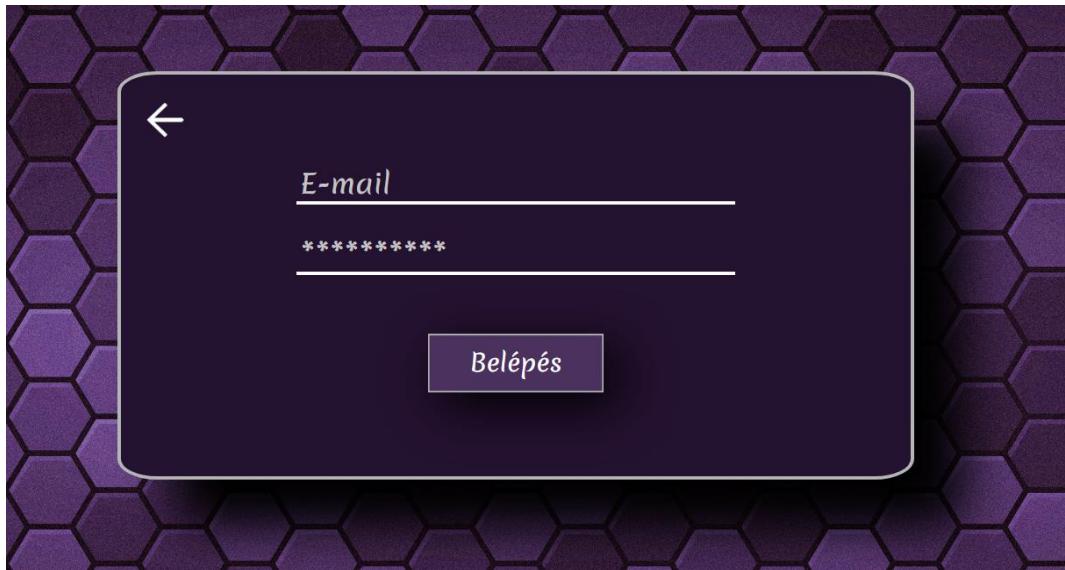
A „Játék Leírása” gombra húzva az egeret, megjelenik egy szövegrész, ahol a játék lényegét, és azzal kapcsolatos információkat olvashatjuk.

Kettő karakter közül a képükre kattintva választhatjuk ki, hogy melyikkel szeretnénk játszani. Csak kiválasztás után lehetséges a névadás. A kiválasztott karakter mérete minimálisan megnő, és világosabb lesz. Egyszerre csak egy karaktert lehet kiválasztva, így ha kiválasztjuk az egyiket, és átkattintunk a másikra, akkor az előző karakter automatikusan elsötétül, míg az újonnan kiválasztott karakter lesz kiemelve. A játék elindításához ki kell választanunk a karakterünket, illetve nevet kell adnunk neki, ahogy a szöveg is említi.

A „Start” gomb megnyomásával elkezdődik a játék.

2.2.2 Bejelentkezés

A bejelentkezés gombra kattintva egy új felületre érkezünk, ahol kettő bemeneti mező jelenik meg: egy e-mail cím és egy jelszó (lásd 3. ábra). Ha rendelkezünk a rendszergazdai belépéshez szükséges adatokkal, akkor zökkenőmentesen be tudunk lépni. Ha hibás adatokkal próbálkozunk meg belépni, akkor hibaüzenetet kapunk, és nem enged belépni.



3. ábra: Bejelentkező felület

2.2.3 Rendszergazda kezdőlap

Bejelentkezés után, a 4. ábrán látható rendszergazdai kezdőlapot láthatjuk. Ugyanúgy, itt is megtalálható a „Start” gomb, amivel el tudjuk indítani a játékot. A „Játék Leírása” gombra húzva az egeret, megjelenik egy szövegrész, ahol az adatbázis szerkesztésével kapcsolatos információkat olvashatjuk. A jobb felső sarokban a „Kijelentkezés” gomb segítségével visszatérhetünk az 1. ábrán látott kezdőlaphoz.



4. ábra: Rendszergazdai kezdőlap

Továbbá látható, hogy három másik gomb vált elérhetővé:

Feladatok szerkesztése (lásd 5. ábra):

A „Feladatok szerkesztése” gombra kattintva egy tábla jelenik meg, ahol lehetőségünk van új adatokat felvenni, szerkeszteni a meglévőket, illetve törölni

azokat. Az új adat felvételéhez három bemeneti mező áll rendelkezésünkre: az elsőben meg kell adnunk a megoldandó feladatot (pl. $1 + 1 =$), a másodikban a karakter által megadott (helytelen) választ a feladatra, a harmadikban pedig a feladat tényleges megoldását. A pluszjelre kattintva az adat automatikusan hozzáadásra kerül a táblához. Az adat szerkesztésére a megfelelő sorban lévő ceruza ikonra kattintva van lehetőségünk. Egy új felületet láthatunk (lásd 6. ábra), ahol a szerkeszteni kívánt adatok megjelennek, és tetszés szerint módosíthatjuk azokat. A pipára kattintva elmentődnek a változtatások, és visszatérünk a Feladatok táblához. Ha törlni szeretnénk sorokat, akkor a megfelelő sorban lévő kuka ikonra kattintva tehetjük meg. A vissza nyíllal újra a rendszergazdai kezdőlapra kerülünk.

Feladat	Válasz	Megoldás	Szerkesztés	Törölés
$1 + 2 =$	12	3		
$3 - 2 =$	32	1		
$2 + 5 =$	25	7		
$10 + 3 =$	103	13		
$3 + 6 =$	36	9		
$1 + 1 =$	11	2		
$2 + 2 =$	22	4		
$5 + 2 =$	52	7		
$3 + 3 =$	33	6		
$5 - 4 =$	2	1		

5. ábra: Feladatok szerkesztése felület

Feladat: $10 + 3 =$

Válasz: 103

Megoldás: 13

6. ábra: Adatszerkesztési felület

Munkahelyek szerkesztése (lásd 7. ábra):

A „Munkahelyek szerkesztése” gombra kattintva egy tábla jelenik meg, ahol lehetőségünk van új adatokat felvenni, és törlni azokat.

Az új adat felvételéhez kettő bemeneti mező áll rendelkezésünkre: az elsőben meg kell adnunk a munka megnevezését, a másodikban pedig a munka nehézségi fokát. A pluszjelre kattintva az adat automatikusan hozzáadásra kerül a táblához. Adatok törlésére a megfelelő sorban lévő kuka ikonra kattintva van lehetőségünk. A vissza nyíllal újra a rendszergazdai kezdőlapra kerülünk.

Az alábbi táblázatban látható a szintek magyarázata (ugyan ez vonatkozik a „Fizetések szerkesztése” pontra).

Szint	Leírása
0	Nincs diplomája
1	Egy kevésbé ismert egyetemről van diplomája
2	Egy átlagosan ismert egyetemről van diplomája
3	A legismertebb egyetemről van diplomája

Megnevezés	Szint	Törlés
Takarító	0	[Delete]
Gondnok	0	[Delete]
Pénztáros	0	[Delete]
Tanár	1	[Delete]
Személyi edző	1	[Delete]
Nővér	1	[Delete]
Informatikus	2	[Delete]
Dietetikus	2	[Delete]
Közgazdász	2	[Delete]
Villamosmérnök	3	[Delete]
Orvos	3	[Delete]
Jogász	3	[Delete]

7. ábra: Munkahelyek szerkesztése felület

Fizetések szerkesztése (lásd 8. ábra):

A „Fizetések szerkesztése” gombra kattintva egy tábla jelenik meg, ahol lehetőségünk van új adatokat felvenni, és törlni azokat.

Az új adat felvételéhez kettő bemeneti mező áll rendelkezésünkre: az elsőben meg kell adnunk a fizetés összegét, a másodikban pedig a fizetés szintjét (megegyezik a munkahelyek nehézségi szintjével). A pluszjelre kattintva az adat automatikusan hozzáadásra kerül a táblához. Adatok törlésére a megfelelő sorban lévő kuka ikonra kattintva van lehetőségünk. A vissza nyíllal újra a rendszergazdai kezdőlapra kerülünk.

Megnevezés	Szint	Törlés
300000	0	☒
350000	0	☒
330000	0	☒
320000	0	☒
310000	0	☒
340000	0	☒
390000	1	☒
360000	1	☒
400000	1	☒
380000	1	☒
370000	1	☒
350000	1	☒
450000	2	☒

8. ábra: Fizetések szerkesztése felület

A táblákban végzett módosítások egyből láthatóak az adatbázisban is.

2.2.4 Játékfelület

2.2.4.1 Bevezetés

A játékban váltakoznak a napszakok (nappal - délután - este), ami a háttér váltakozásával van érzékelhető. A mérföldkövek között eltelt idő (5 év) vizuálisan is látható. Az egyes szakaszok után a karakterünk a korához megfelelő kinézetre vált.

Bármely kezdőlapon a „Start” gombot megnyomva, a 9. ábrán látható felületre érkezünk. A lila gombokkal tudunk navigálni és továbbmenni a játékban.



9. ábra: Játékfelület - Bevezetés

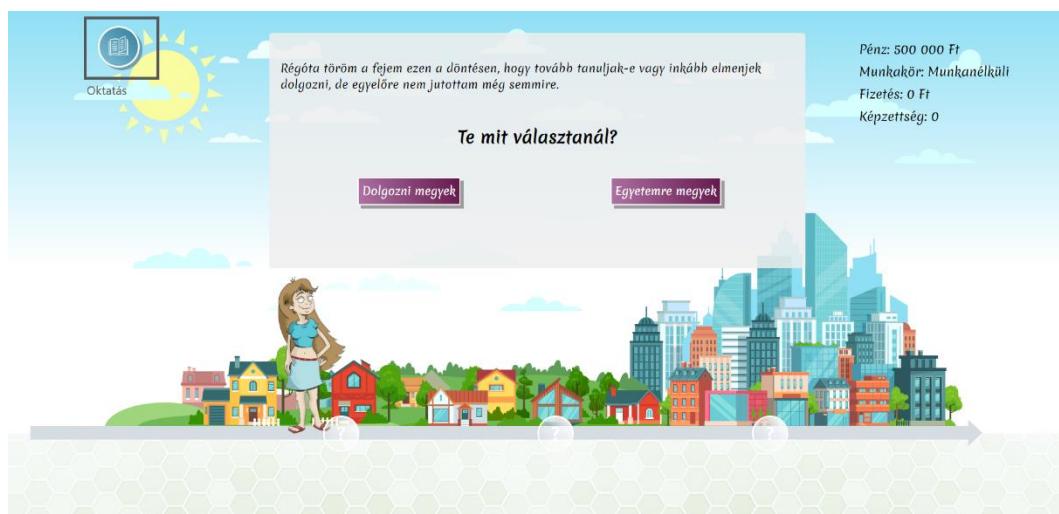
Ha elérkeztünk egy aktív mérföldkőhöz, arra rákattintva jelentkeznek meg a döntési lehetőségeink. A jobb felső sarokban található az állapotjelző, ami a jelenlegi státuszokat mutatja. Egyes szövegrészeken felbukkanak más színű (kék) szavak, amiknek elolvashatjuk a fogalmát az egér ráhúzásakor (lásd 10. ábra).



10. ábra: Játékfelület - Magyarázat

2.2.4.2 I. mérföldkő

Az aktív I. mérföldkőre kattintva a 11. ábrán látható döntési felület jelenik meg, ahol minden kettő lehetőség közül választhatunk. Bárhogyan is döntünk, az befolyással lesz a játék kimenetelére. Véletlenszerű időközönként megjelenik egy gomb kis időre, amivel lehetőségünk van részt venni oktatásokon, hogy képesség pontokra tegyünk szert (maximum 4). Ezek a harmadik mérföldkönél játszanak majd fontos szerepet.



11. ábra: I. mérföldkő

„Dolgozni megyek”:

Miután ráméntünk a „Dolgozni megyek” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. A játék sorsolni fog nekünk egy munkát és egy hozzá tartozó fizetést. Ha tovább megyünk a narráción, kapunk öt feladatot, amik eredményét le kell ellenőriznünk. A játék a feladatokat az adatbázisból kérdezi le és választ ki véletlenszerűen öt darabot. Ezek a feladatok szerkeszthetőek megfelelő jogosultsággal a rendszerüzemi felületen. Ha kijavítottuk az eredményeket és mindegyik válasz helyes, akkor magasabb fizetést kapunk. Ha akár egy feladat megoldása is helytelen, akkor nem kapjuk meg a magasabb fizetést (lásd 12. ábra).



12. ábra: I. mérföldkő – Munka feladatok

Amint elérkeztünk a mérföldkő végéhez, egy összegzést olvashatunk az előző részről, amit a 13. ábrán meg is tekinthetünk. Fel van tüntetve a döntésünk eredménye, a jelenlegi fizetése (bruttó, nettó), illetve az összes bevétele és kiadása 5 év alatt. A bevétel az 5 év alatt kapott nettófizetésünkben van kiszámolva, ez zöld színű szöveggel van jelölve. A kiadás szintén a nettófizetésünkben van kiszámolva, ami pedig piros színnel van jelölve. Mivel csak egy érettségi bizonyítvánnyal rendelkezünk, ezért a 0. szintről indulunk. Ehhez a szinthez tartozik egy alsó és egy felső százalék határ (alsó – 50%, felső – 80%), amik között a játék sorsolni fog egy számot. A fizetésünk sorsolt szám százalékát fogjuk elkölni egy hónapban. A határok minden ugyan azok maradnak, de a szám minden eltelt hónapban változik. Továbbá láthatjuk azt is, hogy már aktív a II. mérföldkő, amivel tovább mehetünk a játékban.



13. ábra: I. mérföldkő - Munka összegzése

„Egyetemre megyek”:

Miután rámentünk az „Egyetemre megyek” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. A felületen egy szerencsekereket láthatunk, amivel kisorsolásra kerül, hogy melyik egyetemre vesznek fel minket a három közül (lásd 14. ábra). A legjobb egyetem a zölddel jelölt rész, a MEW Egyetem, ahova 1/6 az eséllyel lehet bekerülni. A közepesen jó egyetem a sárga színnel jelölt, a VRF Egyetem, ahova már 2/6 az esély a bekerülésre. Végül a legrosszabb egyetem, a KEG Egyetem, ahova 3/6 eséllyel vesznek fel. Minél jobb egyetemre vesznek fel, annál jobb munka ajánlatot és fizetést kapunk a későbbiekk során.



14. ábra: I. mérföldkő - Egyetem sorsolása

A narrációk után elérkezünk a mérföldkő végéhez, ahol egy összegzést olvashatunk az előző részről, amit a 15. ábrán megtekinthetünk. Fel van tüntetve a döntésünk illetve, hogy melyik egyetemre sikerült felvételt nyernünk, a Diákhitel 1 által kapott hitelösszegünk zöld színnel és az egyetemi éveink alatti kiadásunk piros színnel. A kiadásunk 1 hónapban függ attól is, hogy melyik egyetemre vettek fel. A megfelelő egyetemhez rendelt alsó és felső határ között a játék sorsol egy számot, majd 100 000 Ft-nak a sorsolt szám százalékát fogjuk elkölteni. A határok minden eltelt hónapban változik. Az alábbi táblázatban láthatjuk,

hogy melyik szinthez milyen alsó és felső határ tartozik. Továbbá láthatjuk azt is, hogy már aktív a II. mérföldkő, amivel tovább mehetünk a játékban.

Egyetem neve	Szint	Alsó határ	Felső határ
KEG Egyetem	1	60%	80%
VRF Egyetem	2	50%	70%
MEW Egyetem	3	40%	60%



15. ábra: I. mérföldkő - Egyetem összegzése

A kiadásokat és bevételeket az F12 billentyű megnyomásával akár nyomon is tudjuk követni a „Console” felületen (lásd 16. ábra).

```

Console Elements Recorder Performance insights Sources Network >
top | Filter Default
 Hide network  Log XMLHttpRequests
 Preserve log  Eager evaluation
 Selected context only  Autocomplete from history
 Group similar messages in console  Treat code evaluation as user action
 Show CORS errors in console
----- 1. év -----
Keresett összeg 1 év alatt: 2713200
1. hónapban a fizetésének a 63 százalékát költötte el: 142443 Ft
2. hónapban a fizetésének a 79 százalékát költötte el: 178619 Ft
3. hónapban a fizetésének a 59 százalékát költötte el: 133399 Ft
4. hónapban a fizetésének a 52 százalékát költötte el: 117572 Ft
5. hónapban a fizetésének a 57 százalékát költötte el: 128877 Ft
6. hónapban a fizetésének a 72 százalékát költötte el: 162792 Ft
7. hónapban a fizetésének a 49 százalékát költötte el: 110789 Ft
8. hónapban a fizetésének a 63 százalékát költötte el: 142443 Ft
9. hónapban a fizetésének a 59 százalékát költötte el: 133399 Ft
10. hónapban a fizetésének a 67 százalékát költötte el: 151487 Ft
11. hónapban a fizetésének a 66 százalékát költötte el: 149226 Ft
12. hónapban a fizetésének a 67 százalékát költötte el: 151487 Ft
Kiadott összeg 1 év alatt: 1702533 Ft
----- 2. év -----
Keresett összeg 1 év alatt: 2713200
1. hónapban a fizetésének a 71 százalékát költötte el: 160531 Ft
2. hónapban a fizetésének a 66 százalékát költötte el: 149226 Ft
3. hónapban a fizetésének a 68 százalékát költötte el: 153748 Ft
4. hónapban a fizetésének a 62 százalékát költötte el: 140182 Ft
5. hónapban a fizetésének a 53 százalékát költötte el: 119833 Ft
6. hónapban a fizetésének a 76 százalékát költötte el: 171836 Ft
7. hónapban a fizetésének a 54 százalékát költötte el: 122094 Ft

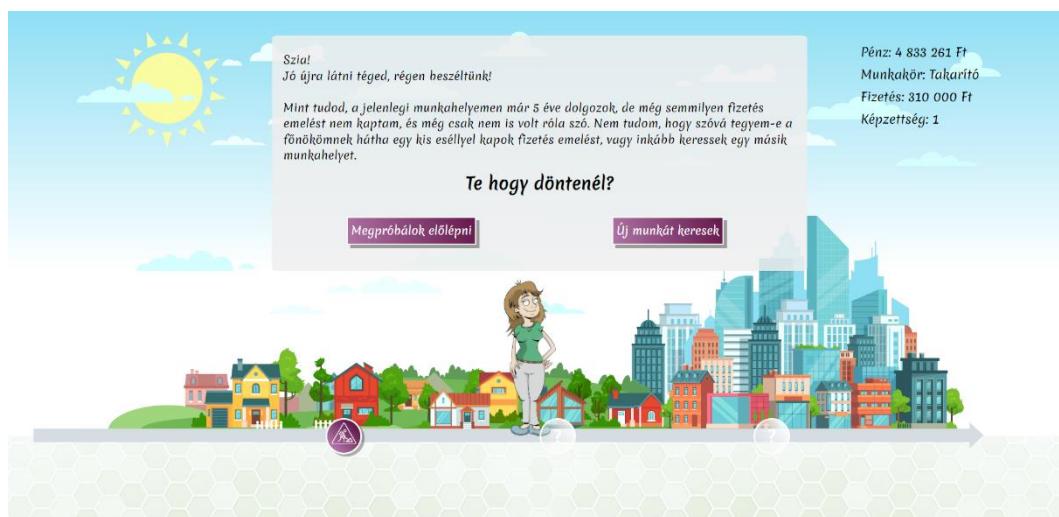
```

16. ábra: Játékelület - Kiadások, bevételek nyomon követése

2.2.4.3 II. mérföldkő

Ha az I. mérföldkönél dolgozni mentünk

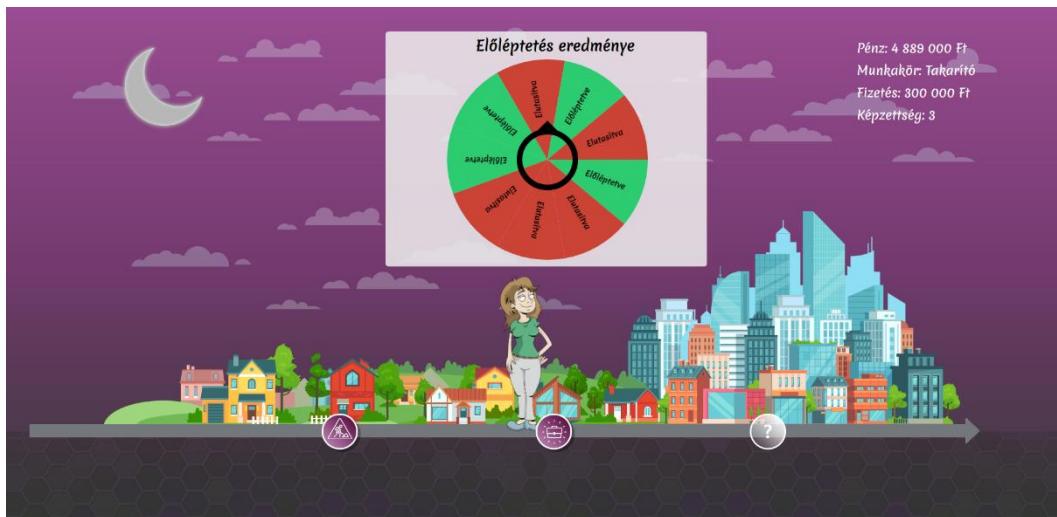
Miután megérkeztünk és rákattintottunk az aktív II. mérföldkőre, a 17. ábrán megtekinthető felületet látjuk. Mivel a két mérföldkő között 5 év telt el, ezért megváltozott a karakterünk kinézete is, sokkal érettebb lett. Továbbra is figyelnünk kell a felbukkanó oktatást jelentő gombokra, hogy minél több képesség pontot gyűjthessünk (maximum 4).



17. ábra: II. mérföldkő – Munkás oldal

„Megpróbálok előlépni”:

Miután rákattintottunk a „Megpróbálok előlépni” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. Ekkor a felületen egy szerencsekereket láthatunk (lásd 18. ábra), ami segítségével ki fogjuk pörgetni, hogy sikerül-e előlépnünk vagy sem. Ha zöld színű „Elöléptetve” mezőre érkezik a nyíl, akkor sikerült előlépnünk, de ez csak 4/9 eséllyel lehetséges. Ennek következményeképp a játék egy magasabb fizetést sorsol számunkra, amit a jobb felső sarokban lévő állapotjelző is mutat. Továbbá szintben is fentebb ugrunk, tehát 0. szintről az 1. szintre. Ellenkező esetben, ha a piros színű „Elutasítva” mezőre érkezik a nyíl, melynek 5/9-hez az esélye, akkor sajnos nem léptetnek elő, a fizetésünk ugyanaz marad, továbbra is a 0. szinten leszünk és a mérföldkő ikonja is megváltozik minimálisan, ami a sikertelen előléptetést jelzi.



18. ábra: II. mérföldkő - Előléptetés eredménye

„Új munkát keresek”:

Miután rákattintottunk az „Új munkát keresek” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. Ekkor a felületen egy narráció fogad, miszerint esélyünk van megszerezni az álom munkánkat. Ha elolvastuk és tovább léptünk a gomb megnyomásával, akkor egy szerencsekereket láthatunk (lásd 19. ábra). Ennek segítségével ki fogjuk pörgetni, hogy megkapjuk-e az álom munkánkat vagy sem. Ha a zöld színű „Felvéve” mezőre érkezik a nyíl, akkor megkapjuk az álom munkát. Ennek az esélye 2/7-hez, így kevészer fordul elő, hogy sikerül megvalósítani az álmunkat. Viszont ha ez éppen sikerül, akkor sokkal magasabb bruttó fizetést kapunk, illetve fentebb is ugrunk 0. szintről a 2. szintre. Ellenkező esetben, ha a piros színű „Elutasítva” mezőre érkezik a nyíl, melynek 5/7-hez az esélye, akkor sajnos nem kapjuk meg az állást. Ettől független sorsolással kapunk egy - az előzőnél színvonalasabb - munkát új bruttó fizetéssel. Ennél a végkimenetelnél is fentebb ugrunk, de a 0. szintről csak az 1. szintre. A sorsolás eredményével a jobb felső sarokban a „Munkakör” és a „Fizetés” is változik, illetve a narráció is említi.



19. ábra: II. mérföldkő - Új munka eredménye

Összegzés:

Miután megkaptuk az eredményt és tovább léptünk a narráción, ahogy a 20. ábrán is látható, egy újabb összegzést kapunk az előző részről. Fel van tüntetve a döntésünk eredménye, a havi bruttó és nettó bérünk, ha változott, akkor az új munkahelyünk, az összes bevételünk zöld színnel és kiadásunk piros színnel jelölve. Ha változott a fizetésünk, akkor a bérدادatoknál a már az újonnan sorsolt összeg jelenik meg, és e szerint is lesz kiszámolva a bevétel illetve a kiadás. Bárhogyan is döntöttünk, minden kettő-kettő végkimenetel van, ezért a kiadások számolásánál az alsó és felső határ más lesz, függeni fog az eredménytől. A bevétel és kiadás kiszámolása a már említett módon történik. Az alábbi tábla szemlélteti a szintekhez tartozó határokat. Továbbá láthatjuk azt is, hogy már aktív a III. mérföldkő, amivel tovább mehetünk a játékban.

Szint	Alsó határ	Felső határ
0	50%	80%
1	40%	70%
2	40%	60%



20. ábra: II. mérföldkő - Munkás oldal összegzése

Ha az I. mérföldkőnél egyetemre mentünk

Rákattintva az aktív II. mérföldkőre és egy kisebb narráció után, a 21. ábrán megtekinthető felületet látjuk. Az eltelt 5 év a karakterünk kinézetén is látszik, sokkal érettebb lett. A játék sorsol számunkra egy munkakört és egy fizetés, amit fent meg is említi a szövegrész. Továbbra is figyelnünk kell a felbukkanó oktatást jelentő gombokra, hogy minél több képesség pontot gyűjthessünk (maximum 4).



21. ábra: II. mérföldkő – Egyetemes oldal

„Elfogadom az állást”:

Miután rákattintottunk az „Elfogadom az állást” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. Ha tovább megyünk a narráción, kapunk öt feladatot, amik eredményét le kell ellenőriznünk. A játék a feladatokat az adatbázisból kérdezi le és választ ki véletlenszerűen öt darabot. Ezek a feladatok szerkeszthetőek megfelelő jogosultsággal a rendszergazdai felületen. Ha kijavítottuk az eredményeket és mindegyik válasz helyes, akkor magasabb fizetést kapunk. Ha akár egy feladat megoldása is helytelen, akkor nem kapjuk meg a magasabb fizetést (lásd 22. ábra).

„Másik állást keresek”:

Miután rákattintottunk az „Másik állást keresek” gombra, a mérföldkő átvált a döntésünknek megfelelő ikonra. Az első állásajánlat óta eltelt 1 év, és nagyon nehezen találunk másik munkát, ezért az első szembejövőt eldöntöttük, hogy elvállaljuk. Azt nem tudni előre, hogy jobb, rosszabb vagy esetleg ugyan olyan lesz-e a fizetésünk, véletlenszerűen sorsolja újra a bérünket a játék. Ha tovább megyünk a narráción, kapunk öt feladatot, amik eredményét le kell ellenőriznünk. A játék a feladatokat az adatbázisból kérdezi le és választ ki véletlenszerűen öt darabot. Ezek a feladatok szerkeszthetőek megfelelő jogosultsággal a rendszergazdai felületen. Ha kijavítottuk az eredményeket és mindegyik válasz helyes, akkor magasabb fizetést kapunk. Ha akár egy feladat megoldása is helytelen, akkor nem kapjuk meg a magasabb fizetést (lásd 22. ábra).

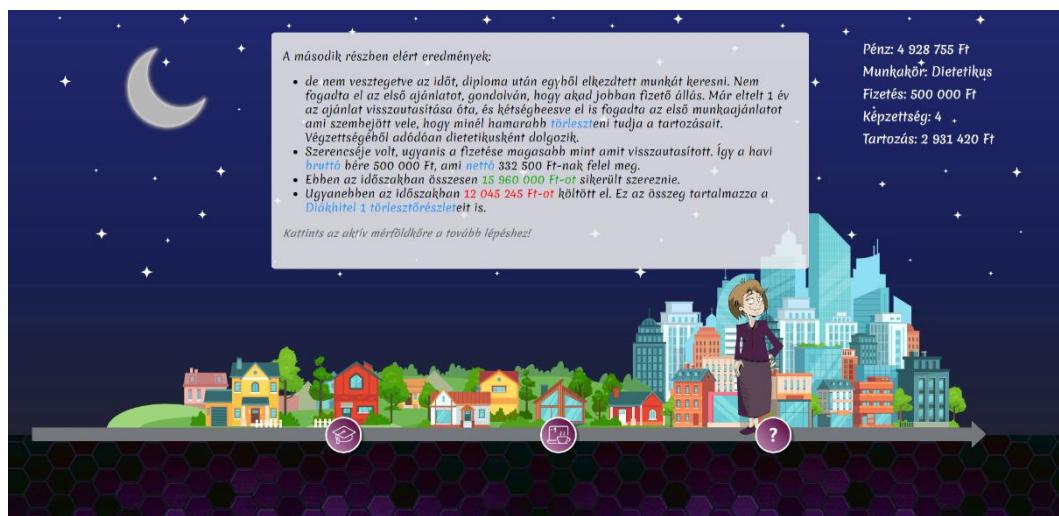


22. ábra: II. mérföldkő - Egyetem utáni munka feladatok

Összegzés:

Amint elérkeztünk a mérföldkő végéhez, egy összegzést olvashatunk az előző részről, amit a 23. ábrán meg is tekinthetünk. Itt bármelyik döntést is választottuk, hasonló az összegzése. Az egyetlen különbség, hogy ha nem fogadtuk el az első állásajánlatot, akkor 1 évig semmilyen bevételünk nem lesz, tehát a bevétel 4 év nettófizetésből áll. Fel van tüntetve a döntésünk eredménye, a jelenlegi fizetésünk (bruttó, nettó), az összes bevételünk zöld színnel jelölve és a kiadásunk piros színnel jelölve. (beletartozik a hitel törlesztése is) 5 év alatt. A bevétel az 5, illetve 4 év alatt kapott nettófizetésünkben van kiszámolva. A bevétel és kiadás kiszámolása a már említett módon történik. A kiadáshoz még beletartozik az egyetemi éveink alatt felvett hitel törlesztőrészlete is. Az alábbi táblázatban szemléltetem a szintekhez tartozó alsó és felső százalék határok. Továbbá láthatjuk azt is, hogy már aktív a II. mérföldkő, amivel tovább mehetünk a játékban.

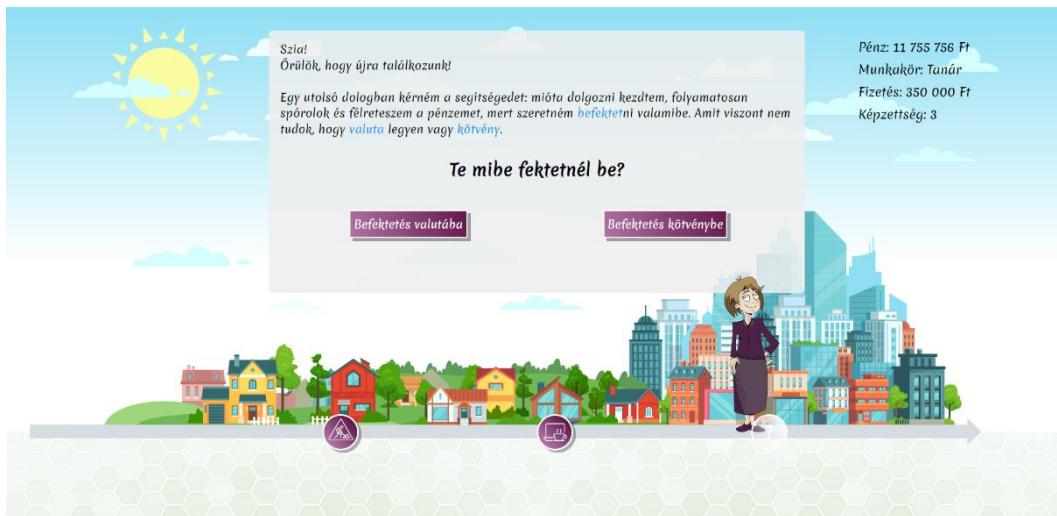
Szint	Alsó határ	Felső határ
1	40%	70%
2	40%	60%
3	40%	50%



23. ábra: II. mérföldkő – Egyetemes oldal összegzése

2.2.4.4 III. mérföldkő

Miután megérkeztünk és rákattintottunk az aktív III. mérföldkőre, a 24. ábrán megtekinthető felületet látjuk. A két mérföldkő között megint 5 év telt el, ezért ismét megváltozott a karakterünk kinézete. Az előzőekben a pénzszerzésen volt a hangsúly, most pedig jöhetsz a befektetések. Bármilyen úton is haladtunk, ennél a résznél már rendelkezünk egy nagyobb megtakarított összeggel. Minél több képességpontunk van, annál szélesebb ismeretséggel rendelkezünk a befektetések terén. Ha 0 vagy 1 képességpontot szereztünk, akkor csak kettő befektetési lehetőségünk van. Ha 2 vagy 3 képességpontot szereztünk, akkor már három lehetőségünk van. Ha viszont megszereztük az összes (4 db) képességpontot, akkor négy lehetőség közül választhatunk a befektetések terén.



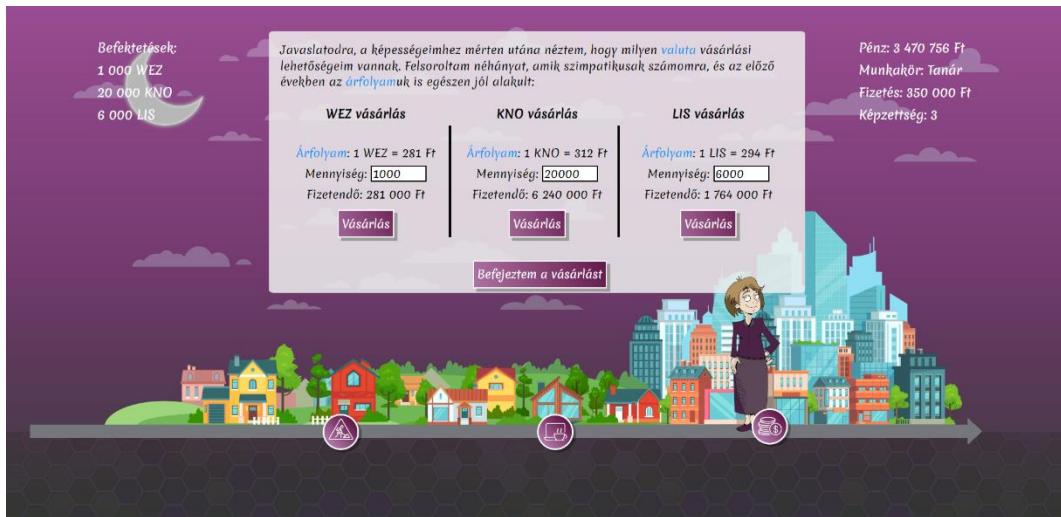
24. ábra: III. mérföldkő

„Befektetés valutába”:

Miután a „Befektetés valutába” gombot választottuk, a mérföldkő átvált a döntésünknek megfelelő ikonra, majd a 25. ábrán látható felület jelenik meg. A lehetőségek mennyisége attól függ, hogy mennyi képesség pontot sikerült szerezni. Látható, hogy a bal felső sarokban megjelent egy „Befektetések” felirat. Ha vásárolunk bármelyik valutából, akkor további feliratok jelennek meg, hogy éppen melyik valutából mennyink van. A kiválasztott valutánál a „Mennyiség” mezőbe be kell írnunk, hogy mennyit szeretnénk vásárolni. A „Fizetendő” felirat mellett kiszámolja az összeget, hogy a beírt mennyiséggű valutáért mennyi Ft-ot kell fizetnünk. Ha akkora összeget szeretnénk vásárolni, amire nincs elegendő pénzünk, nem megy végbe a vásárlás, illetve figyelmeztetve is vagyunk rá. Az ábrán három valutát láthatunk, mindegyik más-más árfolyammal. Az árfolyamok véletlenszerűen sorsolódnak egy

alsó és egy felső határ között. Az alábbi táblázatban látható minden a négy valuta, és annak alsó és felső határai. Ha sikerült befektetni, amibe szerettünk volna, akkor a „Befejeztem a vásárlást” gombra kattintva tovább mehetünk.

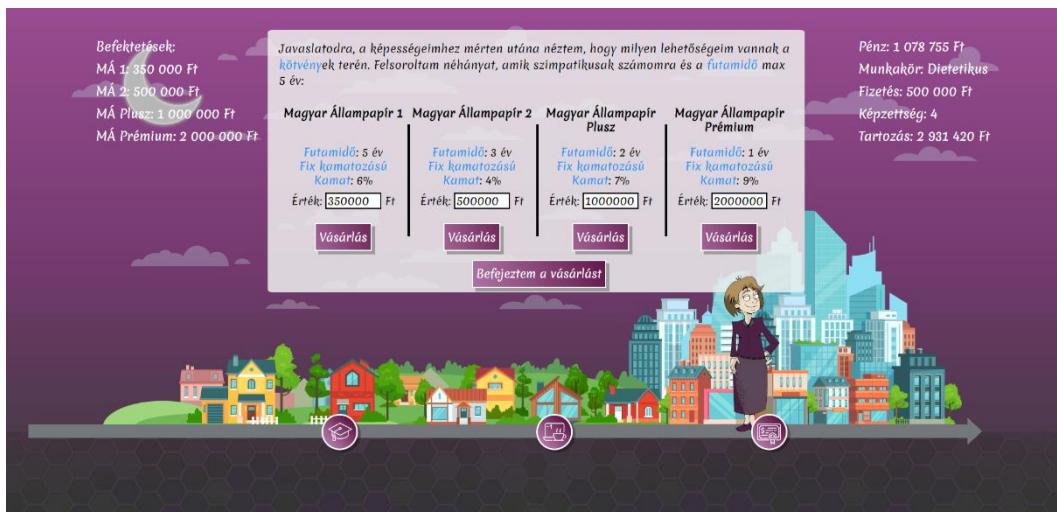
Képzettség	Valuta	Alsó határ	Felső határ
0-1	WEZ	250	350
0-1	KNO	250	350
2-3	LIS	270	330
4	ADP	290	310



25. ábra: III. mérföldkő - Valuta

„Befektetés kötvénybe”:

Miután a „Befektetés kötvénybe” gombot választottuk, a mérföldkő átvált a döntésünknek megfelelő ikonra, majd a 26. ábrán látható felület jelenik meg. A lehetőségek mennyisége attól függ, hogy mennyi képesség pontot sikerült szereznünk. Látható, hogy a bal felső sarokban megjelent egy „Befektetések” felirat. Ha befektetünk bármelyik kötvénybe, akkor további feliratok jelennek meg, hogy éppen milyen értékű állampapíraink vannak. Az ábrán négy kötvényt láthatunk, ezek alatt minden információt megtalálunk, amit tudnunk kell a befektetés során. A kiválasztott kötvénynél az „Érték” mezőbe be kell írnunk egy összeget, amennyiért szeretnénk azt megvásárolni. Ha akkora összeget szeretnénk befektetni, amire nincs elegendő pénzünk, nem megy végbe a vásárlás, illetve figyelmeztetve is vagyunk rá. Ha sikerült befektetni, amibe szerettünk volna, akkor a „Befejeztem a vásárlást” gombra kattintva tovább mehetünk.



26. ábra: III. mérföldkő - Kötvény

Összegzés:

Amint elérkeztünk a mérföldkő végéhez, egy összegzést olvashatunk az előző részről, amit a 27. ábrán meg is tekinthetünk. Itt kettő végkimenetelnek hasonló az összegzsük, egyedül a befektetés módjában tér el. Fel van tüntetve a döntésünk, a befektetéseink, a jelenlegi fizetésünk (bruttó, nettó), az összes bevételünk zöld színnel és a kiadásunk piros színnel jelölve. A befektetéseinknél fel vannak sorsolva, hogy mikbe fektettünk mennyi pénzt, és mennyi kaptunk vissza belőle. Ha valutába fektettünk be, akkor a sorok végén lévő ikonok jelzik, hogy sikerült-e profitálnunk a valuta eladásán. A bal felső sarokban ismét látható a „Befektetések” felirat, ami alatt nem szerepel semmi, mivel a valutákat már eladtuk, illetve a kötvények futamideje is lejárt ilyenkorra. A bevétel az 5 év alatt kapott nettófizetésünk ből és a befektetésből visszafolyt összegből van kiszámolva. A kiadás fő része a nettófizetésünk ből van kiszámolva, de ezen kívül benne vannak a befektetett összegek, és ha az I. mérföldkönél egyetemre mentünk, akkor a maradék hitel törlesztőrészletei is. A munka szintjétől függ, hogy milyen alsó és felső százalék határral számolunk. A megfelelő határ között sorsolni fog egy számot, és a nettófizetésünk sorsolt szám százalékát költjük el egy hónapban. Az alábbi táblázatban szemléltetem a szintekhez tartozó alsó és felső százalék határokat. Legalul látunk egy „Vissza a kezdőlapra” gombot, amit megnyomva a kezdőlapra kerülünk, és akár egy új játékot is kezdhetünk.

Szint	Alsó határ	Felső határ
0	50%	80%
1	40%	70%
2	40%	60%
3	40%	50%



27. ábra: III. mérföldkő - Összegzés

3. Fejlesztői dokumentáció

A következőkben a játék fejlesztői szempontú felépítését mutatom be. A fejezetben található megértést segítő ábrákat és diagramokat a Diagrams.net² segítségével, a drótvázterveket pedig a MockFlow³ nevű webes alkalmazással készítettem el.

3.1 Fejlesztői környezet

Az „Invesztopolisz” egy 1920 x 1080 felbontású, Windows 10-es operációs rendszerű, számítógépre tervezett webes oktató játék.

A frontend implementációja React Js [1] segítségével történt, a megjelenítésért a kézzel írt CSS felel. A backend implementációjáért egyénileg írt PHP felel, az adatok tárolását pedig a Xampp [2] által vezérelt MySQL [3] adatbázis biztosítja. A React (18.2.0) telepítése terminálon keresztül a Node.js-el (v18.14.2), azon belül is az npm (9.5.0) segítségével történt. A fejlesztői környezet a Visual Studio Code.

3.2 Drótváztervek

A játék 3 darab fő nézetből áll: *kezdőlap*, *adatbázis szerkesztő* és *játék felület*.

3.2.1 Kezdőlap

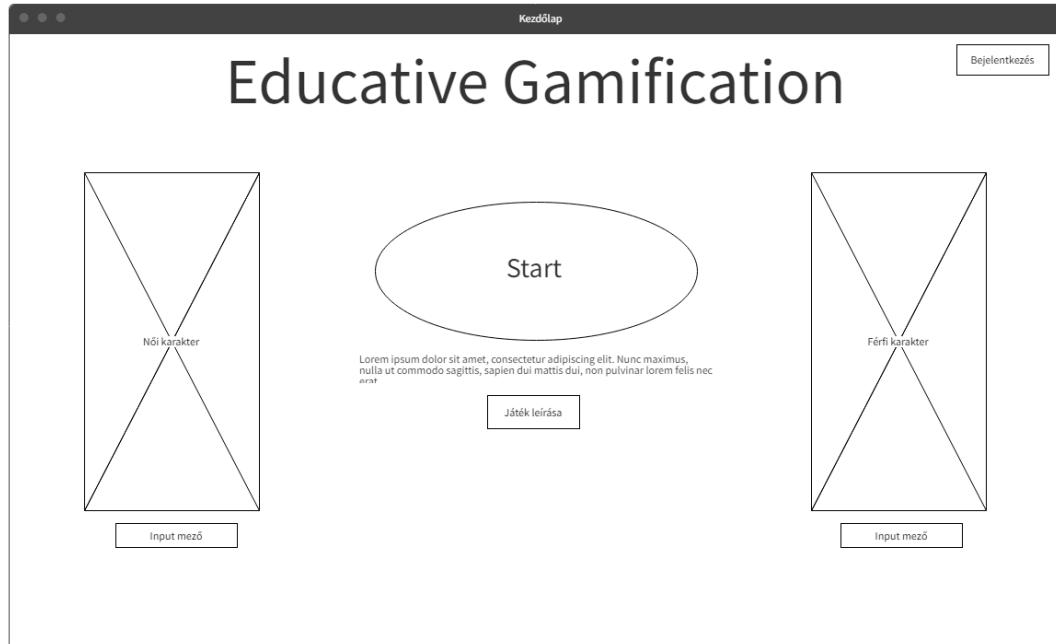
A *kezdőlaphoz* két féle nézet tartozik: *felhasználói kezdőlap*, amit az oldal betöltésekor látunk (lásd 28. ábra), és egy *rendszerelői kezdőlap* (lásd 29. ábra), amit bejelentkezés után érünk el.

Első sorban vegyük a *felhasználói kezdőlapot*. Ahogy látjuk, középen helyezkedik el a cím, jobb felső sarokban a bejelentkezésért felelős gomb. Ezt megnyomva egy bejelentkező felületre érkezünk, ahol a megfelelő adatokkal be tudunk lépni a *rendszerelői kezdőlapra*. A cím alatti rész három oszlobból tevődik össze: bal szélén a női karakter képe, jobb szélén a férfi karakter képe. A középső oszlopból található

² <https://app.diagrams.net>

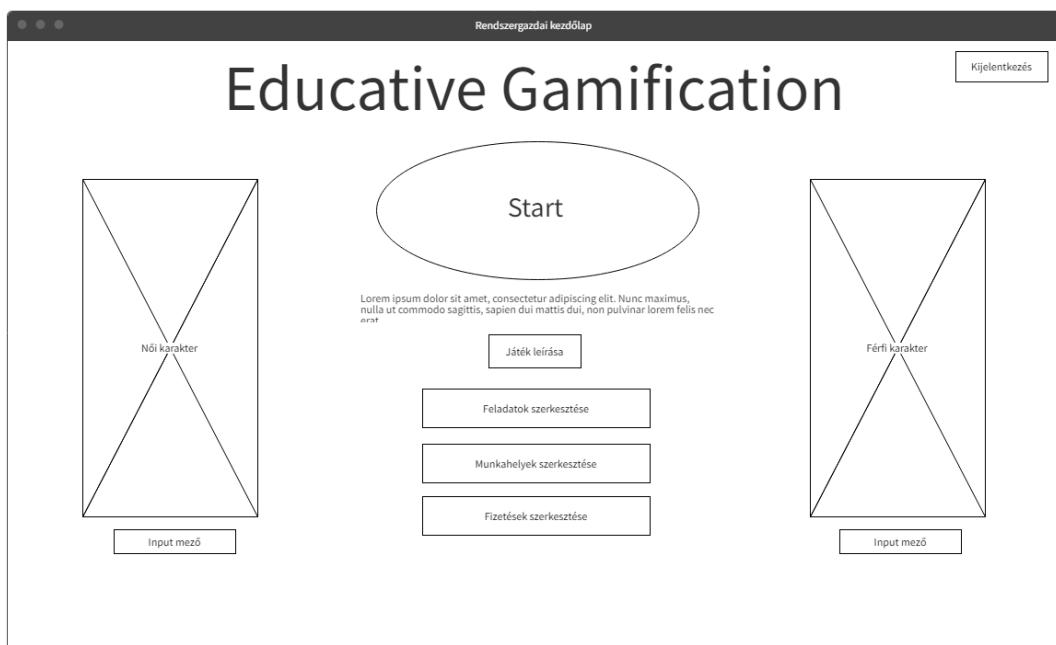
³ <https://www.mockflow.com>

„Start” gomb, ami a játék indításáért felel. Közvetlen ez alatt található egy „Játék leírása” gomb, ami egér ráhúzásakor megjelenít egy szövegrészt.



28. ábra: Drótvázterv - Felhasználói kezdőlap

Bejelentkezés után megjelenik a *rendszerelői kezdőlap*, ahol a *felhasználói kezdőlaphoz* képest a „Játék leírása” gomb alatt plusz három másik gomb jelenik meg. Mindegyik egy *adatbázis szerkesztői felületre* irányít minket. A jobb felső sarokban a „Kijelentkezés” gombot megnyomva visszatérhetünk a *felhasználói kezdőlapra*.



29. ábra: Drótvázterv - Rendszerelői kezdőlap

3.2.2 Adatbázis szerkesztő

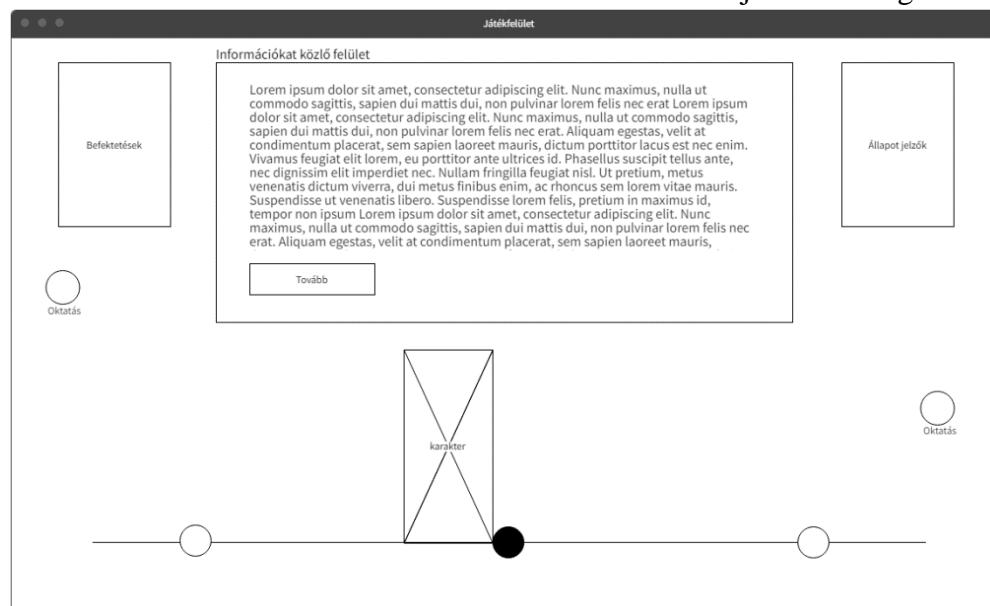
Lehetőségünk van törölni és új adatokat felvenni, illetve a feladatok szerkesztésénél a már meglévő adatokat is tudjuk szerkeszteni (lásd 30. ábra). Középen látható a táblázat, amiben a már adatbázisban szereplő adatok jelennek meg. A táblázat fölött bal oldalt található egy nyíl, ami visszavisz a *rendszerelőlapra*. Az input mezőket ki kell töltenünk azokkal az adatokkal, amit hozzá szeretnénk adni az adatbázishoz. A pluszjelre kattintva hozzáadódik a táblázathoz az újonnan felvett adat.

	Country	Companies
<input type="checkbox"/>	USA	Apple Inc., Microsoft
<input type="checkbox"/>	Sweden	IKEA Furnitures, Spotify
<input checked="" type="checkbox"/>	Finland	Nokia Communications
	Country	Companies
<input checked="" type="checkbox"/>	USA	Apple Inc., Microsoft
<input type="checkbox"/>	Sweden	IKEA Furnitures, Spotify
<input checked="" type="checkbox"/>	Finland	Nokia Communications

30. ábra: Dróttervez - Feladatak szerkesztő felület

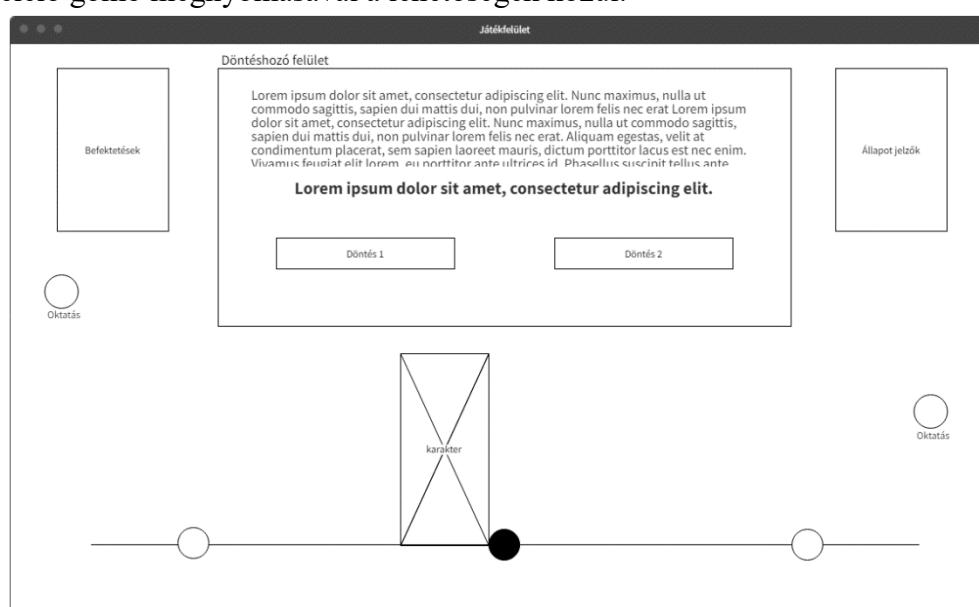
3.2.3 Játékfelület

Bármelyik kezdőlapon a „Start” gomb megnyomása után a 31. ábrán látható *játékfelületre* érkezünk. Középen látható egy szövegdoboz, amiben a játék során folyamatosan narrációs szövegeket olvashatunk. Jobb oldalt az „Állapotjelző” résznél lesz látható a fizetésünk, munkakörünk, pénzzünk, képzettségünk, stb... A bal oldalon a befektetéseink fognak megjelenni. Alul egy idősík helyezkedik el, azon pedig három mérföldkő látható. A karakterünk az idősíkon fog végigmenni, és az éppen aktív mérföldkövekre kattintva a 32. ábrán látható döntési felületek jelennek meg.



31. ábra: Dróttervez - Narrációs játékfelület

A döntések előtt általában egy felvezető szöveget olvashatunk, majd választanunk kell a megfelelő gomb megnyomásával a lehetőségek közül.



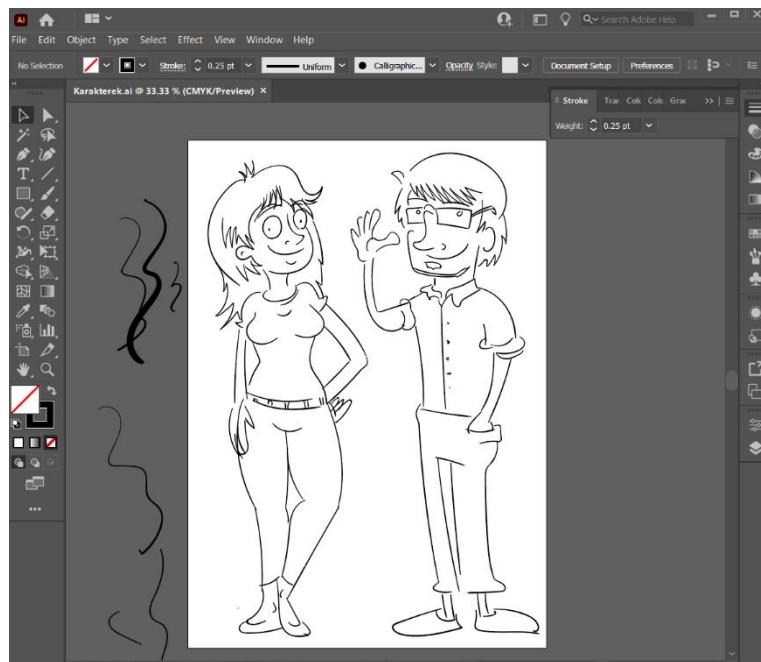
32. ábra: Dróttervez - Döntéshozó felület

3.3 Grafika

A játék során felbukkanó design elemeket (karakterek, hátterek, ikonok) egy kedves ismerősöm, Csorba Gergely készítette el. Néhány kép újra méretezésében, tükrözésében, illetve átszínezésében a Gimp⁴-et vettet segítségül.

3.3.1 Karakterek

A karakterek a WACOM Cintique Pro digitális táblán kézzel lettek rajzolva az Adobe Illustrator⁵ program segítségével (lásd 33. ábra).



33. ábra: Adobe Illustrator program

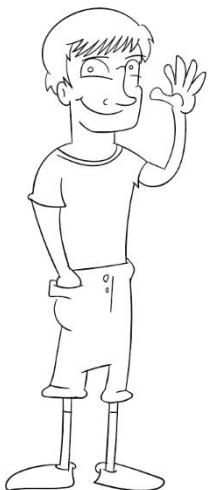
A 34. ábrán a fiatal lány (*young_woman.png*), és a 37. ábrán a fiatal srác (*young_man.png*) kinézete látható, akik most érettségiztek le. Ezt a kinézetet láthatjuk a kezdőlapon, illetve ezzel is fogunk indulni. A 35. ábrán az idősebb női (*older_woman.png*) és a 38. ábrán az idősebb férfi (*older_man.png*) kinézet látható. A második mérföldkőre lépéskor vált át a játék erre a kinézetre és kísér tovább minket. Az előzőhez képest komolyodott, látszik rajta, hogy megszerezte a diplomáját, vagy ha nem egyetemre ment, akkor már van pár év tapasztalata a munkájában. A 36. ábrán látható a magabiztos, érett nő (*old_woman.png*), illetve a 39. ábrán a céltudatos, komoly férfi (*old_man.png*) kinézete. Az utolsó mérföldkőre lépéskor láthatjuk a kinézetet.

⁴ <https://www.gimp.org/>

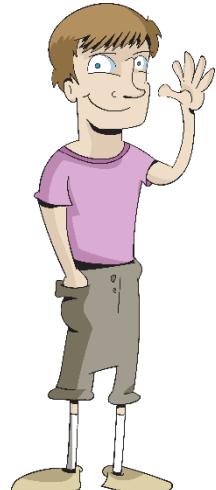
⁵ <https://www.adobe.com/hu/products/illustrator.html>



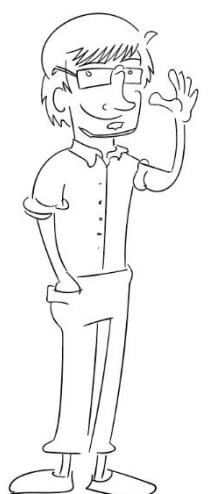
34. ábra: *young_woman.png*



37. ábra: *young_man.png*



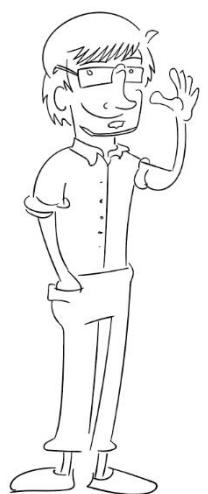
35. ábra: *older_woman.png*



38. ábra: *older_man.png*



36. ábra: *old_woman.png*



39. ábra: *old_man.png*

3.3.2 Hátterek

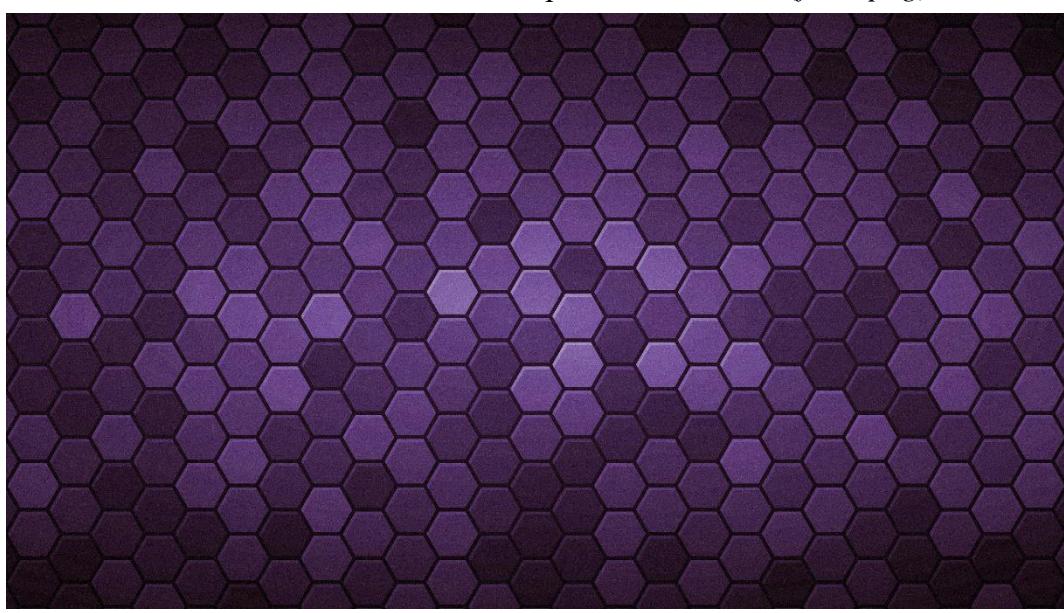
A háttérképek mintája egy ingyenes internetes oldalról, a Freepik⁶-ről lettek letöltve, és a Photoshop⁷ alkalmazással készültek el teljesen. A hexagon mintát szerettem volna belevinni valahogyan, és így születtek meg az alábbi háttérképek:

A 40. ábrán látható kezdőlap háttere (*background.png*)



40. ábra: Kezdőlap háttere

A 41. ábrán látható adatbázis szerkesztő/belépési felület hátttere (*filler.png*)



41. ábra: Adatbázis szerkesztő/belépési felület hátttere

⁶ <http://www.freepik.org/>

⁷ <https://www.adobe.com/hu/products/photoshop.html>

A játék közben három háttérkép váltakozik. A 42. ábrán a nappali háttérkép (*background_day.png*), a 43. ábrán az esti háttérkép (*background_evening.png*), és a 44. ábrán az éjszakai háttérkép (*background_night.png*) látható.



42. ábra: Játékfelület nappali háttérkép



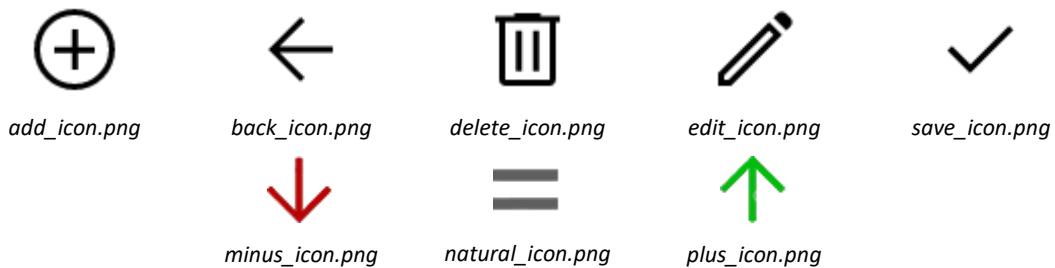
43. ábra: Játékfelület esti háttérkép



44. ábra: Játékfelület éjszakai háttérkép

3.3.3 Ikonok és gombok

A 45. ábrán látható adatbázis szerkesztő felületen lévő ikonokat, illetve a valuta árfolyamát mutató ikonokat a Google Fonts⁸-ról töltöttem le, és a Gimp segítségével méreteztem, illetve színeztem át.



45. ábra: Adatbázis szerkesztő ikonok

A további gombok képei is a Freepik-ról lettek letöltve, és a Photoshop alkalmazással készültek el teljesen. A mérföldköveknek a kettő fő kinézete van: a 46. ábrán látható aktív mérföldkő (*circle_active.png*) és a 47. ábrán látható inaktív mérföldkő (*circle_inactive.png*).



46. ábra:



47. ábra:

⁸ <https://fonts.google.com/icons>

A döntés meghozása után, minden mérföldkő átvált a megfelelő ikonra.

Az 48. ábrán láthatóak az I. mérföldkőhöz tartozó ikonok.



ch1_work.png



ch1_uni.png

48. ábra: I. mérföldkő ikonjai

Az 49. ábrán láthatóak a II. mérföldkőhöz tartozó ikonok.



ch2_promotion.png



ch2_failed_promotion.png



ch2_new_job.png /



ch2_2nd_work_after_uni.png

49. ábra: II. mérföldkő ikonjai

Az 50. ábrán láthatóak a III. mérföldkőhöz tartozó ikonok.



ch3_bonds.png



ch3_valuta.png

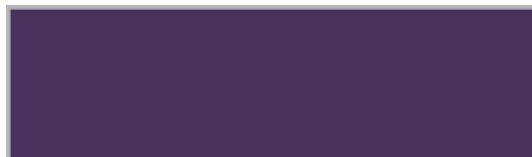
50. ábra: III. mérföldkő ikonjai

A véletlenszerű időközönként felbukkanó oktatás gomb az 51. ábrán látható.



51. ábra: button_skill.png

A játék során kettő féle gombbal találkozhatunk. Az 52. ábrán látható gomb típussal a kezdőlapokon és bejelentkező felületen. Az 53. ábrán látható gomb a játékfelületen található.



52. ábra: button.png

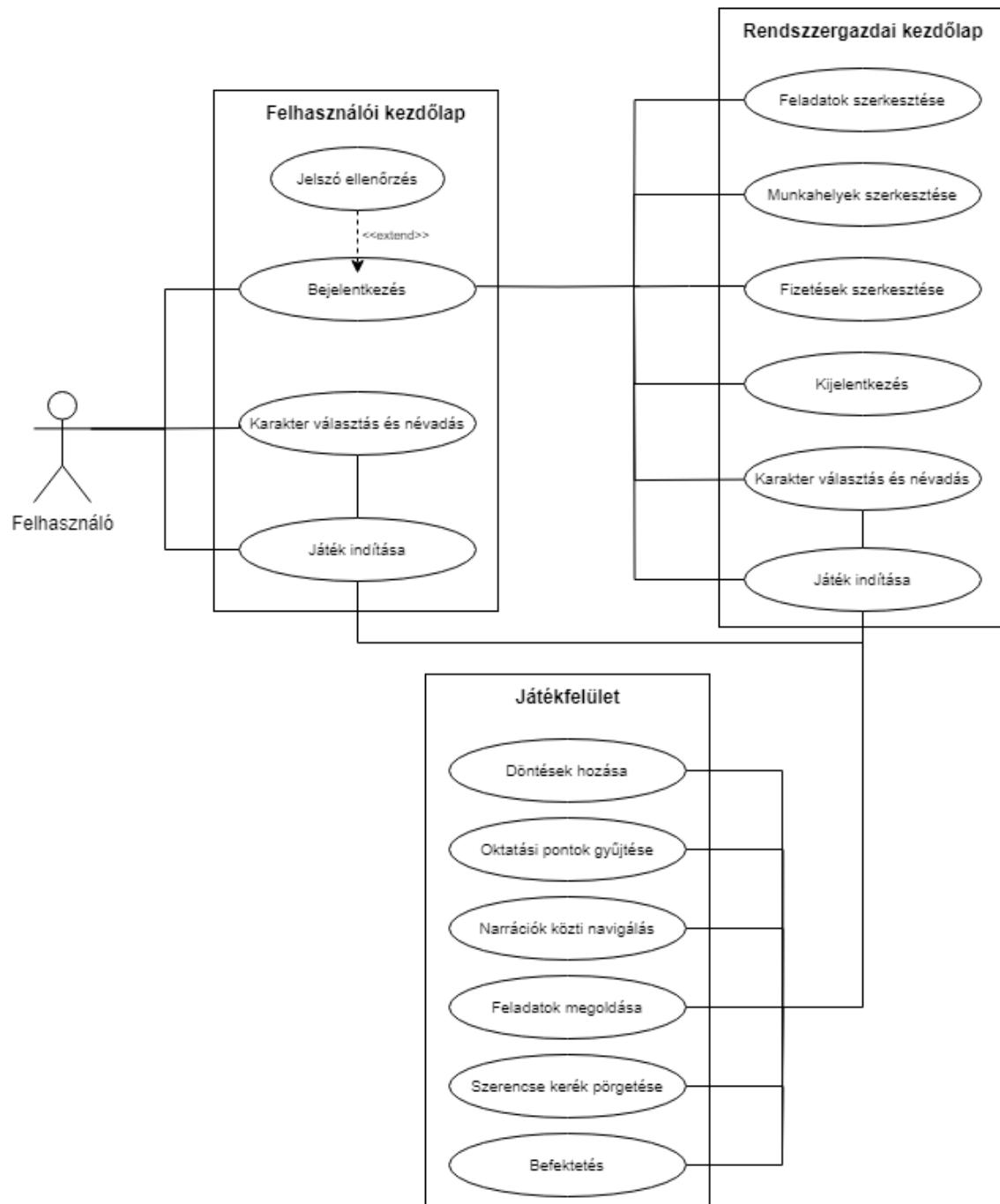


53. ábra: game_button.png

3.4 Használati esetek

A használati esetek 3 fő csoportba sorolhatók, ahogy az 54. ábrán látható:

- *Felhasználói kezdőlap*: Idetartozik a bejelentkezés, karakter választása és játék indítása
- *Rendszergazdai kezdőlap*: Adatbázis szerkesztése, kijelentkezés, karakter választása és játék indítása
- *Játékfelület*: minden, ami történik a játék során

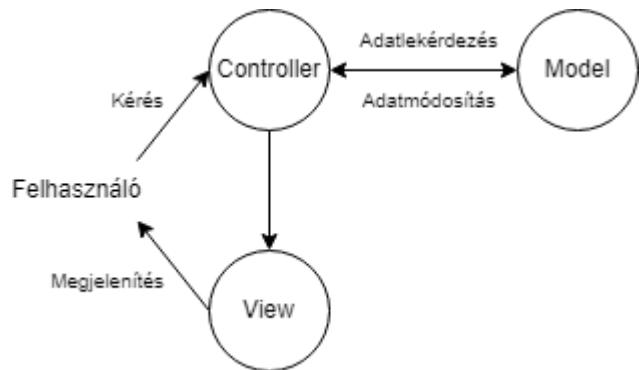


54. ábra: Használati eset diagram

3.5 Architektúra

Az alkalmazás meghatározó architektúrája a *Model-View-Controller* architektúra, ami az 55. ábrán látható.

A *Model* egy valós példány vagy objektum reprezentációja a kódban. A *View* azt a felületet jelenti, amelyen keresztül a felhasználó interakcióba lép az alkalmazással. Amikor egy felhasználó műveletet hajt végre, a *Controller* kezeli a műveletet, és szükség esetén frissíti a *Model*-t.



55. ábra: MVC diagram

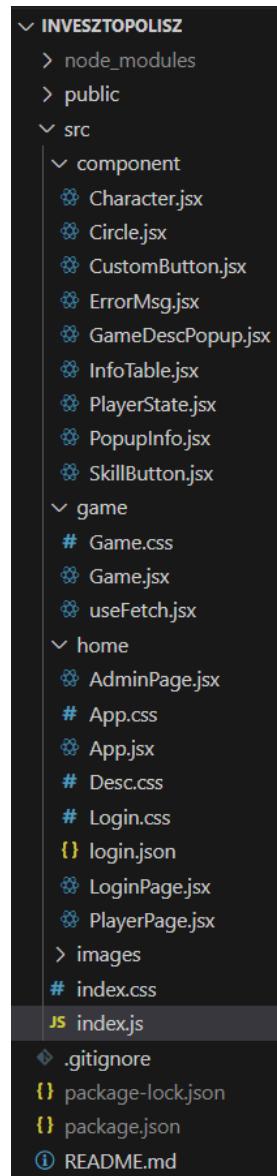
React Js működése

Ha új projektet hoztunk létre, és minden szükséges bővítményt telepítettünk, akkor az 56. és 57. ábrán látható módon épül fel a frontend és a backend mappaszerkezetünk.

Az frontend *src* mappában található *index.js* felel a megjelenítésért (*View*), ez a file rendereli ki és jeleníti meg a kódot vizuális formában a felhasználó felé.

A frontend *home* mappájában látható egy *App.jsx* nevezetű file, amiben definiálva vannak az URL útvonalak. A felhasználó által küldött navigációs kéréseket feldolgozza, majd az alapján a megfelelő oldalra irányít minket. A *PlayerPage.jsx* felel a felhasználói kezdőlapért, ahonnan, ha szeretnénk bejelentkezni, akkor a *LoginPage.jsx* felületre leszünk irányítva. Bejelentkezés után az *AdminPage.jsx* felületre kerülünk, ami felel a rendszergazdai kezdőlapért. A *login.json* fileban vannak tárolva a bejelentkezéshez szükséges adatok. A felhasználói és rendszergazdai kezdőlap kinézetéért az *App.cs*, a bejelentkező ablak kinézetéért pedig a *Login.css* felel.

A frontend *game* mappájában található *Game.jsx* felel a játék működéséért (*Model*), amiben vannak a függvények, a komponens meghívások és adatbázissal kommunikálás. A játékkelület



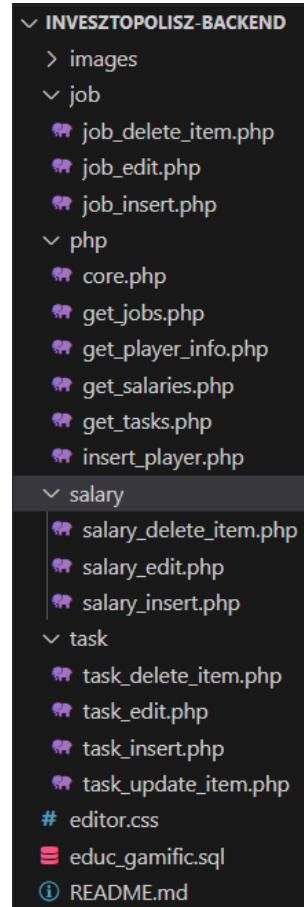
56. ábra: Mappaszerkezet
- Frontend

kinézetéért a *Game.css* file felel. A *useFetch.jsx* file segítségével kérdezhetjük le az adatokat az adatbázisból.

A frontend *component* mappájában találhatóak a modellben meghívott komponensek, melyek kinézetéért szintén a *Game.css* file felel.

A backend *job* mappájában található fileok felelnek a munkahelyek módosításáért, a *salary* mappában lévő fileok a fizetések módosításáért, és a *task* mappa fájljai pedig a feladatok módosításáért felel. Hozzájuk tartozó kinézetért az *editor.css* file felel.

Továbbá a *php* mappa alatt lévő filelek kommunikálnak a modellel és az adatbázissal. Lehetővé teszik, hogy a játéKFelületen lekérjük az adatbázisban tárolt adatokat, illetve onnan beszúrjunk adatot az adatbázisba.



57. ábra: Mappaszerkezet -
Backend

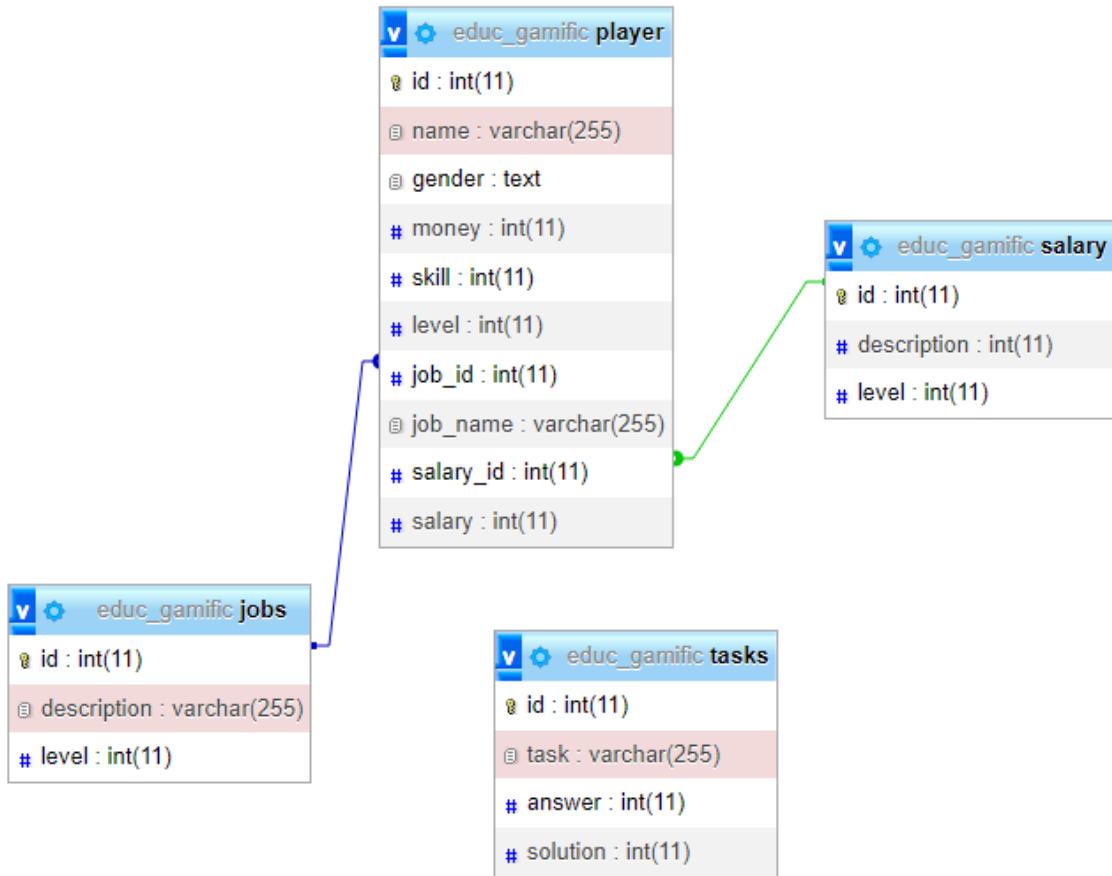
3.6 Adatbázis felépítése

A Xampp által biztosított MySQL adatbázis a 58. ábrán látható módon épül fel. A játék backend része felel az adatbázissal való kommunikálásért.

A játékhoz 4 fő tábla tartozik, amit adatbázisban tárolunk:

- Player: Itt tároljuk a játékoshoz tartozó fontos információkat. A játék indítása előtt beleírjuk a játékos nevét és a kiválasztott karakterét. A tábla az alábbi oszlopokat tartalmazza:
 - id (primary key): játékosokhoz tartozó egyedi azonosító,
 - name: a játékos által megadott karakternév,
 - gender: a karakter neme (n vagy f),
 - money: a játékos pénze,
 - skill: a játékos képességpontjai,
 - level: a játékos szintje, amihez képest kapunk munkát és fizetést,
 - job_id (secondary_key): a kapott munka egyedi azonosítója,
 - job_name: a kapott munka neve,
 - salary_id (secondary_key): a kapott fizetés egyedi azonosítója,
 - salary: a kapott fizetés összege.
- Tasks: Ebben a táblában tároljuk a játék során megoldandó feladatokat. A rendszergazdai kezdőlapon lehetőségünk van módosítani a táblát. A tábla az alábbi oszlopokat tartalmazza:
 - id (primary key): a feladatokhoz tartozó egyedi azonosító,
 - task: a megoldandó feladat,
 - answer: a karakterünk által adott hibás válasz,
 - solution: a feladat helyes megoldása.
- Jobs: Tárolja a lehetséges munkákat, amit a játékos szintjének megfelelően sorsol ki. A tábla az alábbi oszlopokat tartalmazza:
 - id (primary key): a munkákhoz tartozó egyedi azonosító,
 - description: a munka neve,
 - level: a munkához szükséges szint.
- Salary: Tárolja a lehetséges bruttó fizetéseket, amit a játékos szintjének megfelelően sorsol ki. A tábla az alábbi oszlopokat tartalmazza:
 - id (primary key): a fizetéshez tartozó egyedi azonosító,
 - description: a fizetés összege,

- level: a fizetéshez szükséges szint.



58. ábra: Adatbázis felépítése

3.7 Backend

3.7.1 Játékos adatainak beszúrása

Az *AdminPage.jsx* vagy a *PlayerPage.jsx* kezdőlap „Start” gomb megnyomásával lefut az *insertCharacterInfo()* függvény. Ez ellenőrizni fogja, hogy ki van-e választva valamelyik karakter, és el van-e nevezve. Ha igen, akkor meghívja az *insert_player.php* filet a backend oldalról, átadva a játékos nevét és a kiválasztott karaktert. Az *insertCharacterInfo()* függvény az alábbi kódrészletben megtekinthető:

```
const insertCharacterInfo = () => {
  if (isActiveWoman && !isActiveMan && playerName.length > 0) {
    fetch("http://localhost/szakdoga-
backend/php/insert_player.php?player=n&name=" + playerName);
    window.location.href = "http://localhost:3000/game";
  } else if (!isActiveWoman && isActiveMan && playerName.length > 0) {
    fetch("http://localhost/szakdoga-
backend/php/insert_player.php?player=f&name=" + playerName);
    window.location.href = "http://localhost:3000/game";
  }
};
```

59.ábra: *insertCharacterInfo()* függvény

3.7.2 Feladatok szerkesztése

Az *AdminPage.jsx* „Feladatok szerkesztése” gombja a *task_edit.php* felületre irányít át minket. Új adat felvételnél a form a *task_insert.php* file kódját fogja lefutni, ami megtekinthető az alábbi kódrészletben:

```

<?php
    $conection = mysqli_connect('localhost', 'root', '', 'educ_gamific');
    if ($conection->connect_error) {
        die("Connection failed: " . $conection->connect_error);
    }
    $taskName = '';
    $answerNum = '';
    $solutionNum = '';
    if ($_POST){
        $taskName = $_POST['taskName'];
        $answerNum = $_POST['answerNum'];
        $solutionNum = $_POST['solutionNum'];
        $sql_insert = "INSERT INTO `tasks` (`id`, `task`, `answer`, `solution`) VALUES ('0', '$taskName', '$answerNum', '$solutionNum')";
        $rs = $conection->query($sql_insert);
        $conection->close();
    }
    header('Location: task_edit.php');
?>

```

60.ábra: task_insert.php

Első sorban csatlakozunk a megfelelő adatbázishoz, majd a form-ból lekérjük az adatokat a \$_POST metódussal. Létrehozzuk az sql utasítást, amivel beszúrhatjuk az adatokat a táblába, majd végre is hajtjuk az utasítást a \$conection->query(\$sql_insert) parancssal. Végül bezárjuk a kapcsolatot.

A táblázatban a megfelelő sor törlés ikonjára kattintva lefut a *task_delete_item.php* fileban lévő kód, amit meg is tekinthetünk:

```

<?php
    $conection = mysqli_connect('localhost', 'root', '', 'educ_gamific');

    if ($conection->connect_error) {
        die("Connection failed: " . $conection->connect_error);
    }

    $id = $_GET['id'];
    $sql_delete = "DELETE FROM `tasks` WHERE `id`='$id'";
    $result = $conection->query($sql_delete);
    if ($result = 1) {
        $conection->close();
        header('Location: task_edit.php');
    }
    $conection->close();
?>

```

61.ábra: task_delete_item.php

Első sorban kapcsolódunk az adatbázishoz, a `$_GET` metódust használva lekérjük az átadott adatot. Ennek segítségével összeállítjuk az sql parancsot, lefuttatjuk, és bontjuk a kapcsolatot.

A táblázatban lévő adatok szerkesztésére a `task_update_item.php` felületre kerülünk át, ahol hasonlóan a fentebb említett módon megy végbe a folyamat.

3.7.3 Munkák szerkesztése

Az „Munkahelyek szerkesztése” gomb a `job_edit.php` felületre irányít át minket. Új adat felvételéért a `job_insert.php` file felel. A működése megegyezik a feladatoknál lévő új adat felvételével.

A táblázatban a megfelelő sor törléséért a `job_delete_item.php` fileban lévő kód felel. A működése teljesen ugyan az, mint a fentebb említett feladatok törlésénél.

3.7.4 Fizetések szerkesztése

A „Fizetések szerkesztése” gomb megnyomásával a `salary_edit.php` felületre kerülünk. Az új adat beszúrásáért a `salary_insert.php` felel, a törlésért pedig a `salary_edit_item.php`. Ezek működése teljesen ugyanaz, mint a munkahelyeknél.

3.7.5 Egyéb lekérdezések

A backend résznél a `php` mappában az alábbi fileok láthatóak:

- `core.php`: Kezeli a frontend általi lekérdezések hozzáférését a backendhez.
- `get_jobs.php`: A munkák adatbázisból való lekérdezése egy tömbbe, ami az alábbi kódrészletben megtekinthető:

```
<?php
    header('Access-Control-Allow-Origin: *');
    include_once("core.php");
    $connect = mysqli_connect('localhost', 'root',
    '', 'educ_gamific');
    $sql_select = "SELECT * FROM jobs";
    $result = mysqli_query($connect, $sql_select);
    $json_array = array();
    while($row = mysqli_fetch_assoc($result)){
        $json_array[] = $row;
    }
    echo json_encode($json_array);
?>
```

62. ábra: `get_jobs.php`

- *get_player_info.php*: A játékos adatainak adatbázisból való lekérdezése egy tömbbe.
- *get_salaries.php*: A fizetések adatbázisból való lekérdezése egy tömbbe.
- *get_tasks.php*: A feladatok adatbázisból való lekérdezése egy tömbbe.
- *insert_player.php*: Játék kezdése előtt a fontos adatok (név és választott karakter) beszúrása az adatbázis *player* táblájába, ami az alábbi kód részletben megtekinthető:

```
<?php
    header('Access-Control-Allow-Origin: *');
    include_once("core.php");
    $conection = mysqli_connect('localhost', 'root',
'', 'educ_gamific');
    if ($conection->connect_error) {
        die("Connection failed: " . $conection->connect_error);
    }
    $player = $_GET['player'];
    $name = $_GET['name'];
    $sql_delete = "DELETE FROM `player`";
    $result = $conection->query($sql_delete);
    $sql_insert = "INSERT INTO `player` (`name`, `gender`, `money`)
VALUES ('$name', '$player', 500000)";
    $result = $conection->query($sql_insert);
    $conection->close();
?>
```

63.ábra: *insert_player.php*

3.8 Frontend

3.8.1 Felhasználói kezdőlap

A felhasználói kezdőlap működését a PlayerPage.jsx biztosítja, mely függvényei a következők:

- handleClick1(): A női karakterválasztásért felel, ami az alábbi kódrészletben megtekinthető:

```
const handleClick1 = () => {
  if (!isActiveWoman && !isActiveMan) {
    setIsActiveWoman(current => !current);
    setIsActiveMan(current => current);
    setIsActiveWomanInput(current => !current);
    setIsActiveManInput(current => current);
  } else if (isActiveWoman && !isActiveMan) {
    setIsActiveWoman(current => !current);
    setIsActiveMan(current => current);
    setIsActiveWomanInput(current => !current);
    setIsActiveManInput(current => current);
  } else {
    setIsActiveWoman(current => !current);
    setIsActiveMan(current => !current);
    setIsActiveWomanInput(current => !current);
    setIsActiveManInput(current => !current);
  }
};
```

64. ábra: handleClick1() függvény

Megnézi, hogy melyik karakter aktív, és ahhoz képest állítja az ellenkezőjére a karakterek státuszát.

- handleClick2(): A férfi karakterválasztásért felel. Mindkettő karakter lehet egyszerre inaktív, de csak az egyik lehet aktív (kiválasztva).
- setName(e): Az input mező változására beállítja a játékos nevét
- insertCharacterInfo(): A játékos nevét és a kiválasztott karakterét beszúrja az adatbázisba, majd átirányít a játékfelületre.

A return részében látható html kód az, amit át fog adni az App.jsx-nek. Az tovább adja az index.jsx-nek, ami kiértékeli majd megjeleníti a felhasználó számára.

Minden összetartozó rész külön-külön `<div>` tagokban van. A `<div className='mainRow'>`, illetve a `<div className='columns'>` tagok felelnek a 3

oszlopos megjelenítésért, amikben láthatóak a kiválasztható karakterek képei és a „Start” gomb.

3.8.2 Rendszergazdai kezdőlap

A rendszergazdai kezdőlap működését az AdminPage.jsx biztosítja. Ennek működése nagyban megegyezik a felhasználói kezdőlappal. A függvényei teljesen ugyanazok, egyetlen különbség a return html részében van, ugyanis a rendszergazdai kezdőlapnak több kattintható gombja van.

Itt is a `<div className='mainRow'>`, illetve a `<div className='columns'>` tagok felelnek a 3 oszlopos megjelenítésért, amikben láthatóak a kiválasztható karakterek képei és középen a „Start” gomb a többi szerkesztő gombbal.

3.8.3 Belépési felület

Ez köti össze a két kezdőlapot. A felhasználói felületen lévő „Bejelentkezés” gomb megnyomása után bejelentkező felületre leszünk irányítva, amiért a LoginPage.jsx felel. Amint beírtuk a bejelentkezéshez szükséges adatokat és megnyomtuk a „Belépés” gombot, a handleSubmit(e) függvény fog lefutni, ami az alábbi kód részletben megtekinthető:

```
const handleSubmit = (e) => {
    e.preventDefault(); //submit hatására nem törli ki az értékeket
    if (name === data.name && pass === data.pass) {
        setSubmit(true);
        window.location.replace('http://localhost:3000/admin');
    } else {
        setSubmit(false);
    }
};
```

65.ábra: handleSubmit(e) függvény

Összehasonlíta a megadott értékeket a login.json fileban tárolt adatokkal. Ha megegyezik, akkor tovább enged minket a rendszergazdai kezdőlapra, egyébként meg hibaüzenetet fog kiírni, és nem enged tovább.

3.8.4 Útvonalak

Az App.jsx-ben vannak definiálva az útvonalak, hogy milyen kérés esetén melyik oldalra irányítson minket. A <http://localhost:3000> utáni részt vizsgálja, és a szerint adja vissza a megfelelő oldalt.

Ez a program return részében szerepel, és az alábbi kód részletben látható is:

```

function App() {
  return (
    <>
      <BrowserRouter>
        <Routes>
          <Route exact path='/' element={<PlayerPage />}></Route>
          <Route exact path='/login' element={<LoginPage />}></Route>
          <Route exact path='/admin' element={<AdminPage />}></Route>
          <Route exact path='/game' element={<Game />}></Route>
        </Routes>
      </BrowserRouter>
    </>
  );
}

```

66.ábra: App.jsx

Az útvonalak az alábbiak:

- <http://localhost:3000/>: Felhasználói kezdőlap
- <http://localhost:3000/login>: Bejelentkező felület
- <http://localhost:3000/admin>: Rendszergazdai kezdőlap
- <http://localhost:3000/game>: Játékkelület

3.8.5 Komponensek

A komponenseket a modell hívja meg a megfelelő paraméterekkel. A komponensek kinézete/tartalma függ az átadott paraméterektől. A *props.* előtagú változók a paraméterül kapottak. A frontend *component* mappájában az alábbiak találhatóak:

- *Character.jsx*: az átadott paraméternek megfelelő karaktert jeleníti meg a megfelelő kinézzel és módon. Az alábbi kódrészletben látható egy rövid példa:

```

if (props.gender === "?") {
  return <img id='character' className="begin-style" src=''
alt='karakter'></img>
} else if (props.moveTo === 0) {
  if (props.gender === "n") {
    return <img id='character' className="begin-style"
src={youngWoman} alt='karakter'></img>
  } else {
    return <img id='character' className="begin-style"
src={youngMan} alt='karakter'></img>
  }
}
. . .

```

67.ábra: Charater.jsx

- *Circle.jsx*: az átadott paramétereknek megfelelő ikont jeleníti meg a mérföldkövekben az idősíkon. Ez lehet egy aktív vagy inaktív mérföldkő, illetve a különféle döntéseknek megfelelő ikon. Az alábbi kódrészletben láthatjuk az I.mérföldkőhöz tartozó részt, ami eldönti, hogy milyen ikon jelenjen meg:

```

if (props.chapter === 1) {
    if (props.active === true && props.option === 0) {
        return <div id='chapter-one' className='chapter-one-circle-active' onClick={props.onclick}></div>
    } else {
        if (props.option === 0) {
            return <div id='chapter-one' className='chapter-one-circle'></div>
        } else if (props.option === 1) {
            return <div id='chapter-one' className='chapter-one-work'></div>
        } else if (props.option === 2) {
            return <div id='chapter-one' className='chapter-one-uni'></div>
        } else {
            return <></>
        }
    }
}
. . .

```

68.ábra: *Circle.jsx*

- *CustomButton.jsx*: a játékfelületen kattintható gombokat jelenít meg, aminek a felirata az átadott 'chapter' és 'stage' paramétertől függ. Mindig kap paraméterül egy 'onclick' paramétert is, ami megmondja, hogy az adott gomb megnyomásakor melyik függvényt hívja meg. Az alábbi kódrészletben látható rá példa:

```

if (props.chapter === 0 && props.stage === 1) {
    return <button id='to-chapter-one-button'
onClick={props.onclick}>Vágunk bele!</button>
} else if (props.chapter === 1 && props.stage === 14) {
    return <button id='tasks-button'
onClick={props.onclick}>Tovább a feladatokra!</button>
} else if (props.chapter === 1 && props.stage === 15) {
    return <button id='after-tasks-button'
onClick={props.onclick}>Ellenőrzés!</button>
}
. . .

```

69.ábra: *CustomButton.jsx*

- *GameDescPopup.jsx*: A kezdőlapon látott „Játék leírása” szöveget tartalmazza, ami egér ráhúzásakor jelenik meg. A kapott 'place' paraméter határozza meg, hogy a felhasználói vagy a rendszergazdai kezdőlaphoz szükséges szöveget kell megmutatni. Az alábbi kódrészletben látható:

```

if (props.place === "player") {
  return (<>
    <Popup trigger={<button className='popover-button'>Játék
leírása</button>} wide='very'>
      <div className='description'>
        . . .
      </div>
    </Popup>
  </>);
}
. . .

```

70.ábra: *GameDescPopup.jsx*

A *trigger* határozza meg, hogy mi fölött jelenjen meg a szöveg. A *wide* kulcsszóval meghatározható a szövegdoboz mérete. Végül a *<Popup>* tag által közrefogott szövegrész fog majd megjelenni.

- *InfoTable.jsx*: A 'chapter' és 'stage' paraméterek alapján, a játékkelületen középen látható narrációs szövegrész tartalma fog változni. A további átadott paraméterek felhasználásra kerülnek a szövegben. A következő kódrészletben láthatunk erre példát:

```

else if (props.stage === 3) {
  return (
    <div className="chapter-one-table">
      <p className='box0'>Régóta töröm a fejem ezen a
döntésen, hogy tovább tanuljak-e vagy inkább elmenjek dolgozni, de
egyelőre nem jutottam még semmire.</p>
      <div className='box1-ch1'>Te mit választanál?</div>
      <button className='box2-ch1' onClick={e =>
props.onclick1(props.player)}>Dolgozni megyek</button>
      <button className='box3-ch1' onClick={e =>
props.onclick2(props.player)}>Egyetemre megyek</button>
    </div>
  );
}
. . .

```

71.ábra: *InfoTable.jsx*

Függvények a következők:

- handleChange1(event): Az első valutához vagy kötvényhez tartozó mennyiséget állítja be arra, amit beír a játékos az input mezőbe. Az alábbi kódrészletben látható a függvény:

```
const handleChange1 = (event) => {
    setAmount1(event.target.value);
};
```

72.ábra: handleChange1() függvény

- handleChange2(event): A második valutához vagy kötvényhez tartozó mennyiséget állítja be arra, amit beír a játékos az input mezőbe.
- handleChange3(event): A harmadik valutához vagy kötvényhez tartozó mennyiséget állítja be arra, amit beír a játékos az input mezőbe.
- handleChange4(event): A negyedik valutához vagy kötvényhez tartozó mennyiséget állítja be arra, amit beír a játékos az input mezőbe.
- randomNumber(min, max): A paraméterük kapott minimum és maximum határ között generál egy számot, amivel felkerekítés után visszatér. Ez a függvény az alábbi kódrészletben megtekinthető:

```
const randomNumber = (min, max) => {
    return Math.floor(Math.random() * (max - min + 1)) + min;
};
```

73.ábra: randomNumber(min, max) függvény

- *PlayerState.jsx*: Ez teszi láthatóvá a játékfelületen jobb oldalt felül az állapotjelzőket, illetve a befektetési résznél bal felül a különféle befektetéseket. Mindig a legfrissebb értékeket mutatja, paraméterek változásakor ez is folyamatosan frissül és újraértékel. A 'place' paraméter határozza meg, hogy milyen értéket fog mutatni. Az alábbi kódrészletben látható rá egy példa:

```
if (props.place === "money") {
    return <span className={props.class}>Pénz:
{props.value.toLocaleString()} Ft</span>;
} else if (props.place === "job") {
    return <span className={props.class}>Munkakör:
{props.value}</span>;
} else if (props.place === "salary") {
    return <span className={props.class}>Fizetés:
{props.value.toLocaleString()} Ft</span>;
}
. . .
```

74.ábra: PlayerState.jsx

- *PopupInfo.jsx*: A játéKFelületen a különféle narrációkban látható kék színű szavak fogalmát tárolja. A 'place' paraméter határozza meg a szót, és hogy milyen fogalom tartozik hozzá. Példa az alábbi kód részletben látható:

```

if (props.place === "brutto") {
    return <>
        <Popup trigger={<span className='popup-
info'>bruttó</span>} wide>
            <div className='description'>
                Adó és járulék levonás előtti bérösszeg, amit a
munkáltató fizet.
            </div>
        </Popup>
    </>
}
. . .

```

75.ábra: *PopupInfo.jsx*

A *trigger* kulcsszó határozza meg, hogy milyen szó felett jelenjen meg a *<Popup>* tag közt szereplő fogalom. A *wide* kulcsszó a szövegdoboz szélességét határozza meg.

- *SkillButton.jsx*: A véletlenszerűen felbukkanó oktatás gomb megjelenítéséért felel. Az 'active' paraméter határozza meg, hogy látható lehet-e. A 'button' paraméter mondja meg, hogy az első vagy a második gomb jelenjen meg véletlenszerű helyen, amit alábbi kód részletben láthatunk is:

```

if (props.button === 1 && props.active === true) {

    return <div onClick={e => props.onclick(1)}
className='skill-circle' style={{ marginTop: randomNumber(1, 35) +
'%', marginLeft: randomNumber(1, 20) + '%' }}></div>
}
. . .

```

76.ábra: *SkillButton.jsx*

Egyetlen függvénye van, a *randomNumber(min, max)*, ami generál a két határ között egy számot.

3.8.6 Modell

A modell szerepet a *Game.jsx* file tölti be. Ez kezeli a játékmenetet, hív meg minden szükséges komponenst, illetve minden fontos függvény itt van definiálva.

Állapotjelzőért, a befektetésekért, a karakter képéért és a narrációs tábláért felelős függvények:

- setPlayerStateMoney(): Ha már lekérdezte a karakterünk nevét és legalább 0 pénzünk van, akkor megjeleníti az állapotjelző résznél. Ha változik a pénzünk mennyisége, minden változásnál lefut a függvény, így az állapotjelzőnél látható érték mindig frissül vele együtt. A függvény az alábbi kódrészletben látható:

```
const setPlayerStateMoney = () => {
  if (playerInfo.length > 0 && money >= 0) {
    return <PlayerState place="money" value={money}>
  }
  return <></>;
}
```

77.ábra: setPlayerStateMoney() függvény

- setPlayerStateJob(): A karakterünk munkáját jeleníti meg az állapotjelzőnél.
- setPlayerStateSalary(): A karakterünk fizetését jeleníti meg az állapotjelzőnél.
- setPlayerStateSkill(): A karakterünk képzettségét jeleníti meg az állapotjelzőnél.
- setPlayerStateLoan(): A karakterünk tartozását jeleníti meg az állapotjelzőnél.
- setPlayerInvTitle(): Bal oldalt a befektetések címet állítja láthatóvá, ha már döntöttünk a befektetés módjáról.
- setPlayerInv1(): Az első valuta/kötvényből vásárolt összeget állítja.
- setPlayerInv2(): A második valuta/kötvényből vásárolt összeget állítja.
- setPlayerInv3(): A harmadik valuta/kötvényből vásárolt összeget állítja.
- setPlayerInv4(): A negyedik valuta/kötvényből vásárolt összeget állítja.
- getCharacter(): A kezdőlapon kiválasztott karaktert állítja be az alábbi módon:

```
const getCharacter = () => {
  if (playerInfo.length > 0) {
    return <Character gender={playerInfo[0].gender}>
  }
  return <Character gender="?" />;
}
```

78.ábra: getCharacter() függvény

- getInfoTableContent(): A 'chapter' és 'state' változó értékeihez a megfelelő komponenseket hívja meg, ahogy az alábbi kódrészletben látható:

```
const getInfoTableContent = () => {
```

```

        if (playerInfo.length > 0) {
            if (chapter === 0 && stage === 1) {
                return (
                    <div className="begin-game">
                        <InfoTable chapter={0} stage={1}>
                            playername={playerInfo[0].name} />
                            <br />
                            <CustomButton chapter={0} stage={1}>
                                onclick={toCh1Button} />
                            </div>
                );
            } else if . . .
        }
    
```

79.ábra: `getInfoTableContent()` függvény

További segédfüggvények:

- `randomSelector(array, conditionLevel)`: A paraméterül kapott tömbből kiválaszt véletlenszerűen egy elemet, aminek a szintje megfelel a paraméterül kapott szinttel. Ez a függvény az alábbi kód részletben tekinthető meg:

```

function randomSelector(array, conditionLevel) {
    if (array.length > 0) {
        let temp = [];
        let randElem = 0;
        for (let i = 0; i < array.length; i++) {
            if (parseInt(array[i].level) === conditionLevel) {
                temp.push(array[i].id);
            }
        }
        randElem = Math.floor(Math.random() * (temp.length - 1 +
1)) + 1;
        return temp[randElem - 1];
    } else {
        return 0;
    }
}
    
```

80.ábra: `randomSelector(array, conditionLevel)` függvény

- `randomNumber(min, max)`: A paraméterük kapott minimum és maximum határ között generál egy számot, amivel felkerekítés után visszatér.
- `getMultipleRandom(arr)`: A paraméterül kapott tömbből véletlenszerűen kiválaszt öt elemet, majd visszatér a kiválasztott elemekkel.
- `getPaid(player)`: Kiszámolja a játékos nettó fizetéséből az összeget, amit 1 év alatt keresett.

- `spendMoneyWhileWork(player, min, max)`: A paraméterül kapott min és max számok között sorsol egy számot, és a játékos nettó fizetésének ennyi százalékát fogja elkölni egy hónapban. Ez 12 alkalommal fog lefutni, így a sorsolt szám mindenkor más lesz. A havonta kijött eredményeket összeadva megkapjuk, hogy mennyi volt a kiadásunk 1 év alatt. A függvény működését az alábbi kód részletben meg is nézhetjük:

```
function spendMoneyWhileWork(player, min, max) {
    let month = 12; //1 év
    let temp = 0;
    let sum = 0;
    for (let i = 0; i < month; i++) {
        let randomPercent = randomNumber(min, max);
        temp = Math.round((player.salary * 0.665) * (randomPercent / 100));
        sum = sum + temp;
        console.log(i + 1 + ". hónapban a fizetésének a " + randomPercent + " százalékát költötte el: " + temp + " Ft");
    }
    console.log("Kiadott összeg 1 év alatt: " + sum + " Ft");
    return Math.round(sum);
}
```

81.ábra: `spendMoneyWhileWork(player, min, max)` függvény

- `spendMoneyWhileUni(max, min)`: Működése hasonló, mint az előbb említett függvénynél, ugyan úgy a kettő határ között sorsolunk egy számot, viszont itt 100 000 Ft-nak a sorsolt szám százalékát fogja elkölni. Továbbá félévet számolunk év helyett, így a hónapok száma 6.
- `payBackStudentLoan()`: Összeszámolja, hogy mennyit fizetett vissza a hitelből.
- `getSkillLevel (button)`: A paraméterül kapott oktatás gombot megnyomva a láthatóságát hamisra állítja, majd növeli a képzettségi pontjainkat az alábbi módon:

```
const getSkillLevel = (button) => {
    if (button === 1) {
        setSkillB1(false);
    } else {
        setSkillB2(false);
    }
    setPlayerSkill(playerSkill + 1);
    playerInfo[0].skill = playerSkill;
};
```

82.ábra: `getSkillLevel(button)` függvény

Az I. mérföldkőhöz kapcsolódó függvények:

- toCh1Button(): Elindítja az idősíkon a karaktert, majd 5 másodperc elteltével beállítja a pénzünket, a munkánkat, a fizetésünket és aktívvá teszi az I. mérföldkövet.
- setCh1(): Visszaállítja a mérföldkövet inaktívvá, és előhozza a döntési lehetőségeket.

Ha a munkát választottuk:

- getJobCh1(): Beállítja a mérföldkő ikonját a döntésünknek megfelelően, megváltoztatja a hátteret és sorsol nekünk egy munkát, illetve egy hozzá tartozó fizetést.
- beforeTasksCh1(): Narrációt állít.
- getTasks(): Dinamikusan megjeleníti a véletlenszerűen kiválasztott 5 feladatot, amit le kell ellenőriznünk. Az alábbi kódrészletben megtekinthető a függvény:

```
const getTasks = () => { //feladatok kiosztása - CH1 / CH2
    if (tasks.length > 0) {
        let temp = tempArray.map((task, index) => (<div
key={index}>{index + 1}.| {task.task}<input type='text' id={index}
defaultValue={task.answer} className='task-input'
onChange={checkAnswer} autoComplete="off"></input></div>));
        return (
            <div className='table-narration'>
                <div className='task-title'>A feladatok a
                    következők: </div>
                    {temp}
                    <CustomButton chapter={1} stage={15}
                    onclick={increaseMoneyAfterTasks} />
            </div>
        );
    } else {
        return (
            <div className='table-narration'>
                <div className='task-title'>Nincsen
                    megjeleníthető feladat!</div>
                    <CustomButton chapter={1} stage={15}
                    onclick={increaseMoneyAfterTasks} />
            </div>
        );
    }
};
```

83.ábra: getTasks() függvény

- checkAnswer(e): Helyes válasz esetén növeli a pontszámunkat.
- increaseMoneyAfterTasks(): Beállítja a következő hátteret, megadja a magasabb bruttó alapfizetést ha hibátlanul teljesítettük a feladatokat, majd tovább megyünk a narrációval.
- backFromWorkCh1(): Kiszámolja, hogy 5 év alatt mennyit kerestünk, illetve mennyit költöttünk el, majd ennek megfelelően beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívrá állítja a mérföldkövet.

Ha az egyetemet választottuk:

- startUniCh1(): Narrációt állít.
- spinWheelCh1(): Beállítja a hátteret, és előhozza a pörgethető szerencsekereket.
- afterSpinWheelCh1(winner): Kiértékeli a pörgetett eredményt, és annak megfelelően állítja be a játékos szintjét. Továbbá módosítja a hátteret, beállítja a munkakört, hozzáadja a pénzéhez a felvett hitel összegét és beállítja a tartozás állapotjelzőjét.
- backFromUniCh1(): Kiszámolja, hogy 10 félév alatt mennyit költöttünk el az egyetemen, majd ennek megfelelően beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívrá állítja a mérföldkövet.

A II. mérföldkőhöz kapcsolódó függvények:

- setCh2(): Visszaállítja a mérföldkövet inaktívvá, újra nappali hátteret állít be, ha egyetemre mentünk az I. mérföldkönél, akkor módosítja a munkakörünket és előhozza a következő döntési lehetőségeket.

Ha munka után az előléptetést választottuk:

- spinWheelPromotionCh2(): Beállítja a hátteret, a mérföldkő ikonját a döntésünknek megfelelően, majd előhozza a pörgethető szerencsekereket.
- afterSpinWheelPromotionCh2(winner): Beállítja a következő napszakot és kiértékeli a pörgetés eredményét. Sikeres előléptetés esetén magasabbra állítja a játékos szintjét, és ennek megfelelően új fizetést is sorsol.
- backFromSuccessPromotionCh2() és backFromFailedPromotionCh2(): Kiszámolja, hogy 5 év alatt mennyit kerestünk, illetve mennyit költöttünk el a

szintünknek megfelelően, majd beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívra állítja a következő mérföldkövet.

Ha munka után az új munkát választottuk:

- newJobCh2(): Beállítja a hátteret és a mérföldkő ikonját a döntésünknek megfelelően.
- spinWheelNewJobCh2(): Beállítja a pörgethető szerencsekereket.
- afterSpinWheelNewJobCh2(): Beállítja a következő napszakot és kiértékeli a pörgetés eredményét. Sikeres álom munka megszerzése esetén kettővel magasabba állítja a játékos szintjét, és ennek megfelelően új munkát és fizetést is sorsol. Ellenkezőleg csak egyel magasabba állítja a játékos szintjét, majd új munkát és fizetést sorsol.
- backFromSuccessNewJobCh2() és backFromFailedNewJobCh2(): Kiszámolja, hogy 5 év alatt mennyit kerestünk, illetve mennyit költöttünk el a szintünknek megfelelően, majd ennek beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívra állítja a következő mérföldkövet.

Ha egyetemre mentünk:

- jobAfterUniCh2(): Napszakot állít, az egyetemi szintünkhoz képest sorsol egy munkát és egy fizetést, amiről dönthetünk, hogy elfogadjuk-e vagy sem.
- acceptJobCh2() és newJobAfterUniCh2(): Beállítja a mérföldkő ikonját a döntésünknek megfelelően. Ha elfogadtuk az első ajánlatot, akkor beállítja a munkát és a fizetést. Ha nem fogadtuk el az első ajánlatot, akkor sorsol egy új fizetést, és beállítja az értékeket.
- beforeTasksCh2(): Narrációt állít. Ez után jön a feladatmegoldó rész, ami fentebb már említésre került: (→ getTasks() → checkAnswer(e) → increaseMoneyAfterTasks()).
- backFromWorkFirstJobCh2(): Kiszámolja, hogy 5 év alatt mennyit kerestünk, illetve mennyit költöttünk el a szintünknek megfelelően, a hitel törlesztését is beleszámolva, majd beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívra állítja a következő mérföldkövet.
- backFromWorkSecondJobCh2(): Kiszámolja, hogy 4 év alatt mennyit kerestünk, illetve 5 év alatt mennyit költöttünk el a szintünknek megfelelően, a hitel

törlesztését is beleszámolva, majd beállítja a pénzünket. Elindítja a karakterünket a következő mérföldkőhöz, és ha odaért, aktívra állítja a következő mérföldkövet.

A III. mérföldkőhöz kapcsolódó függvények:

- setCh3(): Visszaállítja a mérföldkövet inaktívvá, újra nappali hátteret állít be és megjeleníti a döntési lehetőségeket.

Ha valutába fektettünk be:

- selectCurrenciesInvestmentCh3(): Beállítja a következő hátteret és a mérföldkő ikonját a döntésünknek megfelelően.
- selectValuta1Ch3(amount): Ha vásárolunk az első fajta valutából és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt mennyiségű valutát a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre. Ez a függvény a következő kód részletben megtekinthető:

```
const selectValuta1Ch3 = (amount) => {
    if (amount > 0) {
        if (parseInt(playerInfo[0].money) - (parseInt(amount) * val1) >= 0) {
            playerInfo[0].money = parseInt(playerInfo[0].money)
            - (parseInt(amount) * val1);
            setMoney(playerInfo[0].money);
            setInv1(inv1 + parseInt(amount));
        } else {
            setErrorMsg(true);
            setTimeout(function () {
                setErrorMsg(false);
            }, 3000);
        }
    }
};
```

84.ábra: selectValuta1Ch3(amount) függvény

- selectValuta2Ch3(amount): Ha vásárolunk a második fajta valutából és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt mennyiségű valutát a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.
- selectValuta3Ch3(amount): Ha vásárolunk a harmadik fajta valutából és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt

mennyiséggű valutát a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.

- `selectValuta4Ch3(amount)`: Ha vásárolunk a negyedik fajta valutából és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt mennyiséggű valutát a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.

Ha kötvénybe fektettünk be:

- `selectBondsInvestmentCh3()`: Beállítja a következő háttérét és a mérföldkő ikonját a döntésünknek megfelelően.
- `selectBond1Ch3(amount)`: Ha vásárolunk az első fajta kötvényből és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt értékű kötvényt a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre, ami az alábbi kódrészletben látható is:

```
const selectBond1Ch3 = (amount) => {
    if (amount > 0) {
        if (parseInt(playerInfo[0].money) - parseInt(amount) >= 0) {
            playerInfo[0].money = parseInt(playerInfo[0].money)
            - parseInt(amount);
            setMoney(playerInfo[0].money);
            setInv1(inv1 + parseInt(amount));
        } else {
            setErrorMsg(true);
            setTimeout(function () {
                setErrorMsg(false);
            }, 3000);
        }
    }
};
```

85. ábra: `selectBond1Ch3(amount)` függvény

- `selectBond2Ch3(amount)`: Ha vásárolunk a második fajta kötvényből és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt értékű kötvényt a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.
- `selectBond3Ch3(amount)`: Ha vásárolunk a harmadik fajta kötvényből és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt értékű

kötvényt a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.

- `selectBond4Ch3(amount)`: Ha vásárolunk a negyedik fajta kötvényből és van rá pénzünk, akkor levonja az összeget a pénzünkből, és hozzáadja a vásárolt értékű kötvényt a bal oldali befektetések részhez. Ha vásároltunk, de nincs rá elegendő pénzünk, akkor egy üzenetet fog megjeleníteni 3 másodpercre.

Kettő összegzése:

- `summaryOfInvestmentCh3()`: Összegezi a befektetéseinket egy narrációs részben.
- `getInvestmentCh3()`: Kiszámolja, hogy 5 év alatt mennyit kerestünk, illetve mennyit költöttünk el a szintünknek megfelelően, majd beállítja a pénzünket. Ha az I. mérföldkönél egyetemre mentünk, akkor a maradék hitel törlesztése is hozzáadódik a kiadásokhoz. Ha kötvénybe fektettünk be, akkor a vásárolt kötvények megfelelő futamidő lejárata után hozzáadja a pénzünkhöz a kamatozott összeget. Ha valutába fektettünk be, akkor 5 év után az új árfolyamon eladja a valutákat, és az összeget hozzáadja a pénzünkhöz.
- `backToMenu()`: Visszavisz a felhasználói kezdőlapra.

3.9 Bővítési lehetőségek

Az Invesztopolisz webes játékot számtalan módon lehetne még bővíteni. Az alábbiakban leírom azokat a terveket, amiket szívesen megvalósítottam volna, ha több idő áll rendelkezésre.

Karakterek személyre szabása

Jelenleg egy női és egy férfi karakter közül lehet választani, amik előre meg vannak rajzolva. Sokkal élvezetesebb lenne a játék, ha mi készíthetnénk el a karakterünket. Több testalkatból és haj típusból lehetne választani. Továbbá külön lehetne színezni a haját, a szeme színét, vagy akár a bőre színét is.

További mérföldkövek hozzáadása

A játék jelenlegi állásában 3 mérföldkő van. További mérföldkövek hozzáadásával meghosszabbodna a játékmenet, és még több befejezés lenne. Pl.: szerencsejáték (lottázás, póker, stb...), további befektetések (részvény, ingatlan, stb...)

Életcélok bevezetése

Így egy történetet és egy befejezést adhatunk a karakterünknek, hiszen egész játék alatt arra koncentrálnánk és a szerint cselekednénk, hogy a karakterünk elérje a célját. A játék elején sorsolva kaphatnánk meg az életcélt. Néhány példa: „Gyűjts össze 30.000.000 Ft-ot, majd vegyél belőle egy házat.”, „Nyerd meg a lottó főnyereményét!”, „Fektess be legalább 10.000.000 Ft-ot.”, stb...

Véletlenszerű események bővítése

Sokkal eseménydúsabb lehetne a játék, pozitív vagy akár negatív események is történhetnének. Negatív esemény például jelenthet több pénzkiadást egészségügyi okokból, lefokozást, munka elvesztését, stb... Pozitív esemény lehet például egy kihagyhatatlan munkaadási ajánlat, pénzt találunk a földön, hősies tettek, stb...

Szerkeszthető játékmenet

A játékban előre megadott lehetőségek vannak, amik közül dönthetünk. A rendszergazdai felületen lehetne egy gomb, ami egy szerkesztő felületre vinne minket. Itt tudnánk új döntéseket hozzáadni a játékhoz, illetve szerkeszteni a már meglévőket, amit elmentve lehetőségünk lenne kijátszani. Továbbá az elmentett játékmenetet vissza lehetne tölteni, és tovább szerkeszteni a döntéseket ha szükséges.

3.10 Tesztelés

A játék tesztelése manuális tesztek segítségével valósult meg, ami elsősorban a felület működését ellenőrzi.

3.10.1 Manuális tesztek

Az alábbiakban részekre elkülönítve találhatóak meg a tesztek a leírásukkal együtt:

Felhasználói kezdőlap:

- *Bejelentkezés:* A gombra kattintva egy új felületet láthatunk, megjelenik egy fehér vissza nyíl, és kettő kitölthető mező, ahol meg kell adnunk egy e-mail címet és egy jelszót.
- *Játék leírása:* Az egeret ráhúzva egy fehér szövegdoboz jelenik meg, ami tartalmazza a játék leírását.
- *Karakterek:*
 - Ha egyik karakter sincs kiválasztva, akkor egy tetszőleges karakterre kattintva a karakter mérete nagyobb lesz, színe világosabb lesz, és tölthető lesz az alatta lévő kitölthető mező.
 - Ha átkattintunk a másik karakterre, akkor az előző elsötétül, visszamegy az eredeti méretére és nem lesz tölthető az alatta lévő mező, míg az újonnan kiválasztott karakter mérete megnő, kivilágosodik, és az alatta lévő input mező lesz tölthető.
 - Ha már egy kiválasztott karakterre kattintunk rá, akkor az elsötétül és visszamegy az eredeti méretébe, tehát egyik karakter sem lesz kiválasztva éppen. így egyik bemeneti mező sem lesz tölthető.
- *Start:*
 - Ha kiválasztottuk a karakterünket és adtunk neki nevet, akkor a gombra kattintva egy játékfelületet láthatunk. Itt láthatjuk az általunk kiválasztott karaktert és a narrációban olvashatjuk a nevét.
 - Ha nem választottunk ki karaktert, vagy kiválasztottuk de nem adtunk neki nevet, akkor nem indul el a játék, nem fog semmi se történni.

Bejelentkező felület:

- *Vissza nyíl:* A nyíl megnyomása után visszatérünk a felhasználói kezdőlapra.

- Belépés:
 - Sikeres belépés esetén a rendszergazdai kezdőlapra érkezünk, ahol a felhasználói kezdőlaphoz képes plusz három másik gombot láthatunk a „Játék leírása” gomb alatt
 - Sikertelen belépés esetén kapunk egy hibaüzenetet, és nem enged tovább minket.

Rendszergazdai felület:

- *Kijelentkezés*: A gombra kattintva visszatérünk a felhasználói kezdőlapra.
- *Játék leírása*: Az egeret ráhúzva egy fehér szövegdoboz jelenik meg, ami tartalmazza az adatbázis módosításához szükséges információkat.
- *Karakterek*:
 - Ha egyik karakter sincs kiválasztva, akkor egy tetszőleges karakterre kattintva a karakter mérete nagyobb lesz, színe világosabb lesz, és tölthető lesz az alatta lévő kitölthető mező.
 - Ha átkattintunk a másik karakterre, akkor az előző elsötétül, visszamegy az eredeti méretére és nem lesz tölthető az alatta lévő mező, míg az újonnan kiválasztott karakter mérete megnő, kivilágosodik, és az alatta lévő input mező lesz tölthető.
 - Ha már egy kiválasztott karakterre kattintunk rá, akkor az elsötétül és visszamegy az eredeti méretébe, tehát egyik karakter sem lesz kiválasztva éppen. így egyik bemeneti mező sem lesz tölthető.
- *Start*:
 - Ha kiválasztottuk a karakterünket és adtunk neki nevet, akkor a gombra kattintva egy játékkelületet láthatunk. Itt láthatjuk az általunk kiválasztott karaktert és a narrációban olvashatjuk a nevét.
 - Ha nem választottunk ki karaktert, vagy kiválasztottuk de nem adtunk neki nevet, akkor nem indul el a játék, nem fog semmi se történni.
- *Feladatok szerkesztése*: A gombra kattintva a játék adatbázis szerkesztő felületére érkezünk, ahol a feladatokat módosíthatjuk.
- *Munkahelyek szerkesztése*: A gombra kattintva a játék adatbázis szerkesztő felületére érkezünk, ahol a munkahelyeket módosíthatjuk.
- *Fizetések szerkesztése*: A gombra kattintva a játék adatbázis szerkesztő felületére érkezünk, ahol a fizetéseket módosíthatjuk.

Feladatok szerkesztése felület:

- *Vissza nyíl*: A nyíl megnyomása után visszatérünk a rendszergazdai kezdőlapra.
- *Hozzáadás*: A pluszjel ikonra kattintva hozzáadhatunk egy új sort a középen látható táblázathoz. A tőle balra lévő három bemeneti mezőbe írhatjuk a hozzáadni kívánt adatokat.
- *Szerkesztés*: Bármelyik sor ceruza ikonjára kattintva, egy új felületet láthatunk, ahol a kiválasztott sor adatai láthatóak.
- *Törlés*: Bármelyik sor kuka ikonjára kattintva, a kiválasztott sor törlődni fog, nem lesz többé a táblázatban látható.

Munkahelyek szerkesztése felület:

- *Vissza nyíl*: A nyíl megnyomása után visszatérünk a rendszergazdai kezdőlapra.
- *Hozzáadás*: A pluszjel ikonra kattintva hozzáadhatunk egy új sort a középen látható táblázathoz. A tőle balra lévő kettő bemeneti mezőbe írhatjuk a hozzáadni kívánt adatokat.
- *Törlés*: Bármelyik sor kuka ikonjára kattintva, a kiválasztott sor törlődni fog, nem lesz többé a táblázatban látható.

Fizetések szerkesztése felület:

- *Vissza nyíl*: A nyíl megnyomása után visszatérünk a rendszergazdai kezdőlapra.
- *Hozzáadás*: A pluszjel ikonra kattintva hozzáadhatunk egy új sort a középen látható táblázathoz. A tőle balra lévő kettő bemeneti mezőbe írhatjuk a hozzáadni kívánt adatokat.
- *Törlés*: Bármelyik sor kuka ikonjára kattintva, a kiválasztott sor törlődni fog, nem lesz többé a táblázatban látható.

Játékfelület:

Bevezetés

- „*Vágunk bele*”: A gomb megnyomása után a karakterünk elindul az I. mérföldkőhöz, a tábla szövege megváltozik. Amint megérkezik, a mérföldkő aktívrá változik, és megjelenik a „Pénz”, „Munkakör”, „Fizetés” és „Képzettség” felirat a jobb felső sarokban.
- *Kék szöveg*: Egér ráhúzásakor megjelenik a szó jelentése.

- *Aktív I. mérföldkő*: Kattintáskor a mérföldkő visszaváltozik inaktívra, megváltozik a tábla szövege, megjelenít kettő gombot a narráció alatt, amik a választható döntési lehetőségek.
- *Kék oktatás gomb (1)*: Miközben olvassuk a szöveget, megjelenik az első szerezhető oktatás gomb a baloldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.

I. mérföldkő – Munka

- „*Dolgozni megyek*”: A gomb megnyomásakor a háttér átvált délutánra, megváltozik a narrációs tábla szövege, a mérföldkő ikonja átvált a munkához megfelelő ikonra, a „Munkakör” és a „Fizetés” státuszok változnak arra a munkára és fizetésre, amit a játék sorsolt ki nekünk.
- *Kék oktatás gomb (2)*: Miközben olvassuk a szöveget, megjelenik a második szerezhető oktatás gomb a jobboldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.
- „*Tovább a feladatokra*”: A gomb megnyomásakor ismét megváltozik a tábla. Öt feladatot láthatunk felsorolva egymás alatt, és egy gombot.
- „*Ellenőrzés*”: A gomb megnyomásakor változik a tábla szövege, a háttér estére vált és ellenőrzi az eredményeket. Ha hibátlan lett, akkor változik a „Fizetés” státusz.
- *Kék oktatás gomb (3)*: Miközben olvassuk a szöveget, megjelenik a harmadik szerezhető oktatás gomb a baloldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.
- „*Még beszélünk*”: A gomb megnyomása után a karakterünk elindul a II. mérföldkőhöz és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak és bevételünknek megfelelően. Amint megérkezik, a II. mérföldkő aktívra vált és megváltozik a karakterünk kinézete.

I. mérföldkő – Egyetem

- „*Egyetemre megyek*”: A gomb megnyomásakor megváltozik a tábla szövege, a mérföldkő ikonja átvált az egyetemnek megfelelő ikonra.
- *Kék oktatás gomb (2)*: Miközben olvassuk a szöveget, megjelenik a második szerezhető oktatás gomb a jobboldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.

- „*Eredmények megtekintése*”: A gomb megnyomása után egy szerencsekereket láthatunk középen a narrációs táblában, illetve a háttérkép átvált délutánra.
- *Szerencsekerek*: Kattintáskor a kerék elkezd forogni, és véletlenszerűen megáll valahol. Vár három másodpercet, majd megváltozik a tábla tartalma, a háttér átvált estére, módosul a „Munkakör” státusz egyetemistára és a „Pénz” státusz a hitel miatt. Továbbá a jobb felső sarokban, a „Képzettség” alatt megjelenik egy új státusz a felvett diákhitel miatt, a „Tartozás”.
- *Kék oktatás gomb (3)*: Miközben olvassuk a szöveget, megjelenik a harmadik szerezhető oktatás gomb a baloldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.
- „*Még beszélünk*”: A gomb megnyomása után a karakterünk elindul a II. mérföldkőhöz és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak megfelelően. Amint megérkezik, a II. mérföldkő aktívra vált és megváltozik a karakterünk kinézete.

II. mérföldkő – Munka

- *Aktív II. mérföldkő*: Kattintáskor a mérföldkő visszaváltozik inaktívra, a háttér nappalra vált, megváltozik a tábla szövege, megjelenít kettő gombot a narráció alatt, amik a választható döntési lehetőségek.

II. mérföldkő – Munka – Előléptetés

- „*Megpróbálok előlépni*”: A gomb megnyomása után egy szerencsekereket láthatunk középen a narrációs táblában, illetve a háttérkép átvált délutánra. A mérföldkő ikonja pedig át fog váltani az előléptetésnek megfelelő ikonra.
- *Szerencsekerek*: Kattintáskor a kerék elkezd forogni, és véletlenszerűen megáll valahol. Vár három másodpercet, majd megváltozik a tábla tartalma és a háttér átvált estére. Ha sikeresen előléptünk, akkor módosul a „Pénz” státusz az újonnan sorsolt fizetésre. Ha nem sikerült előlépnünk, akkor a mérföldkő ikonja átvált a sikertelen előléptetést jelző ikonra.
- *Kék oktatás gomb (4)*: Miközben olvassuk a szöveget, megjelenik a negyedik szerezhető oktatás gomb a jobboldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.
- „*Még találkozunk*”: A gomb megnyomása után a karakterünk elindul a III. mérföldkőhöz és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik

a kiadásunknak megfelelően. Amint megérkezik, a III. mérföldkő aktívra vált és megváltozik a karakterünk kinézete.

II. mérföldkő – Munka – Új munka

- „*Új munkát keresek*”: A gomb megnyomásakor megváltozik a tábla szövege egy narrációra, a háttérkép átvált délutánra, mérföldkő ikonja pedig át fog váltani az új munkakeresésnek megfelelő ikonra.
- „*Döntés megtekintése*”: A gomb megnyomása után egy szerencsekereket láthatunk középen a narrációs táblában.
- *Szerencsekerék*: Kattintáskor a kerék elkezd forogni, és véletlenszerűen megáll valahol. Vár három másodpercet, majd megváltozik a tábla tartalma és a háttér átvált estére. Az eredménytől függetlenül kapni fogunk egy új munkát és fizetést, amire át fog váltani a jobb felső sarokban lévő „Munkakör” és „Fizetés” státusz.
- *Kék oktatás gomb (4)*: Miközben olvassuk a szöveget, megjelenik a negyedik szerezhető oktatás gomb a jobboldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.
- „*Még találkozunk*”: A gomb megnyomása után a karakterünk elindul a III. mérföldkőhöz és a tábla tartalma átvált egy összegzsre. A „Pénz” státusz változik a kiadásunknak megfelelően. Amint megérkezik, a III. mérföldkő aktívra vált és megváltozik a karakterünk kinézete.

II. mérföldkő – Egyetem

- *Aktív II. mérföldkő*: A mérföldkőre kattintáskor a háttér nappalra vált, és megváltozik a tábla tartalma.
- „*Tovább*”: A gomb megnyomása után a háttérkép délutánra vált át és ismét megváltozik a tábla tartalma. Felül egy narrációt olvashatunk, alatta pedig kettő gombot láthatunk, amik a döntési lehetőségeink.
- *Kék oktatás gomb (4)*: Miközben olvassuk a szöveget, megjelenik a negyedik szerezhető oktatás gomb a jobboldalon valahol. Gombra kattintáskor egyből eltűnik a gomb, és a jobb felső sarokban látható „Képzettség” pontunk nő eggyel.

II. mérföldkő – Egyetem – Első állás

- „*Elfogadom az állást*”: A gomb megnyomása után a narrációs rész megváltozik, a mérföldkő ikonja átvált a döntésnek megfelelő ikonra. A „Munkakör” és a „Fizetés” státusz megváltozik arra, amit a játék sorsolt számunkra.

- „*Tovább a feladatokra*”: A gomb megnyomásakor ismét megváltozik a tábla. Öt feladatot láthatunk felsorolva egymás alatt, és egy gombot.
- „*Ellenőrzés*”: A gomb megnyomásakor változik a tábla szövege, a háttér estére vált és ellenőrzi az eredményeket. Ha hibátlan lett, akkor változik a „Fizetés” státusz.
- „*Még találkozunk*”: A gomb megnyomása után a karakterünk elindul a III. mérföldkőhöz és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak és bevételünknek megfelelően, illetve a „Tartozás” státusz a felére csökken. Amint megérkezik, a III. mérföldkő aktívrá vált és megváltozik a karakterünk kinézete.

II. mérföldkő – Egyetem – Második állás

- „Másik állást keresek”: A gomb megnyomása után a narrációs rész megváltozik, a mérföldkő ikonja átvált a döntésnek megfelelő ikonra. A „Munkakör” és a „Fizetés” státusz megváltozik arra, amit a játék sorsolt számunkra (a fizetést itt kétszer sorsolja újra, mivel az elsőt nem fogadtuk el).
- „*Tovább a feladatokra*”: A gomb megnyomásakor ismét megváltozik a tábla. Öt feladatot láthatunk felsorolva egymás alatt, és egy gombot.
- „*Ellenőrzés*”: A gomb megnyomásakor változik a tábla szövege, a háttér estére vált és ellenőrzi az eredményeket. Ha hibátlan lett, akkor változik a „Fizetés” státusz.
- „*Még találkozunk*”: A gomb megnyomása után a karakterünk elindul a III. mérföldkőhöz és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak és bevételünknek megfelelően, illetve a „Tartozás” státusz a felére csökken. Amint megérkezik, a III. mérföldkő aktívrá vált és megváltozik a karakterünk kinézete.

III. mérföldkő

- *Aktív III. mérföldkő*: Kattintáskor a mérföldkő visszaváltozik inaktívrá, a háttér nappalra vált, megváltozik a tábla szövege, megjelenít kettő gombot a narráció alatt, amik a választható döntési lehetőségek.

III. mérföldkő – Valuta

- „Befektetés valutába”: A gomb megnyomása után a mérföldkő ikonja átvált a döntésnek megfelelőre, a háttér délutánra vált és a narrációs rész is megváltozik, oszlopos kinézetet kap, ami függ a „Képzettség” státusztól. Bal felső sarokban megjelent egy új „Befektetések” státusz.
- „Vásárlás”: Miután a felette lévő mezőbe írtunk mennyiséget, a gombot megnyomva, a mennyiséget felszorozza a megfelelő árfolyammal, és e szerint fog változni a „Pénz” státusz. Ha nincs elég pénzünk a vásárlásra, a karakter mellett három másodpercre megjelenik egy szöveg. Sikeres vásárlás után a beírt mennyiségű valuta megjelenik bal felül a „Befektetések” státusz alatt.
- „Befejeztem a vásárlást”: A gomb megnyomásakor változik a tábla szövege és a háttér estére vált.
- „Még látjuk egymást”: A gomb megnyomása után a karakterünk elindul a játék végéhez és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak és bevételünknek megfelelően. Ha egyetemre mentünk, akkor a „Tartozás” státusz el fog tűnni teljesen és a bal oldali részen lévő „Befektetések” státusz alatti részek is eltűnnek.
- „Vissza a kezdőlapra”: A gomb megnyomása után a felhasználói kezdőlapra kerülünk.

III. mérföldkő – Kötvény

- „Befektetés kötvénybe”: A gomb megnyomása után a mérföldkő ikonja átvált a döntésnek megfelelőre, a háttér délutánra vált és a narrációs rész is megváltozik, oszlopos kinézetet kap, ami függ a „Képzettség” státusztól. Bal felső sarokban megjelent egy új „Befektetések” státusz.
- „Vásárlás”: Miután a felette lévő mezőbe írtunk értéket, a gombot megnyomva változni fog a „Pénz” státusz a beírt mennyiségnek megfelelően. Ha nincs elég pénzünk a vásárlásra, a karakter mellett három másodpercre megjelenik egy szöveg. Sikeres vásárlás után a beírt értékű kötvény megjelenik bal felül a „Befektetések” státusz alatt.
- „Befejeztem a vásárlást”: A gomb megnyomásakor változik a tábla szövege és a háttér estére vált.

- „Még látjuk egymást”: A gomb megnyomása után a karakterünk elindul a játék végéhez és a tábla tartalma átvált egy összegzésre. A „Pénz” státusz változik a kiadásunknak és bevételünknek megfelelően. Ha egyetemre mentünk, akkor a „Tartozás” státusz el fog tűnni teljesen és a bal oldali részen lévő „Befektetések” státusz alatti részek is eltűnnek.
- „Vissza a kezdőlapra”: A gomb megnyomása után a felhasználói kezdőlapra kerülünk.

A fentebb kifejtett tesztek az alábbi eszközökön lettek futtatva:

- Asus Rog Strix G15 Laptop:
 - processzor: Intel Core i7-10750H 2.60 GHz
 - memória: 16 GB
 - rendszer típusa: 64 bites operációs rendszer
 - operációs rendszer: Windows 10
- Asztali számítógép:
 - processzor: Intel Core i7-8700K 3.7 GHz
 - memória: 32 GB
 - rendszer típusa: 64 bites operációs rendszer
 - operációs rendszer: Windows 10

4. Forrásjegyzet

[1] „React Js Documentation” [Online]. Link: <https://legacy.reactjs.org/docs/getting-started.html>

Elérés dátuma: 2023.05.10.

[2] „Xampp Documentation” [Online]. Link: <https://www.apachefriends.org/hosting.html>

Elérés dátuma: 2023.03.04.

[3] „MySQL Documentation” [Online]. Link: <https://dev.mysql.com/doc/>

Elérés dátuma: 2023.03.29.