

Pécsi Szakképzési Centrum Simonyi Károly Szakgimnáziuma  
és Szakközépiskolája

*Szakképesítés neve: Szoftverfejlesztő - OKJ száma: 54 213 05*

# **Záródolgozat**

Nemzetek - online stratégia, szerepjáték

Készítette: Rózsa Zoltán Dávid

**Pécs**

**2021**

## **NYILATKOZAT**

Alulírott Rózsa Zoltán Dávid, kijelentem, hogy jelen szakdolgozatom saját munkám eredménye, a felhasznált irodalmat és forrásokat a követelménynek megfelelően feltüntettem.

Kelt: Pécs, 2021. április 12.

.....

Aláírás

# Tartalomjegyzék

|         |  |    |
|---------|--|----|
| 1.      | Előszó.....  | 4  |
| 1.1     | A téma ismertetése .....                           | 5  |
| 1.2     | Köszönetnyilvánítás .....                          | 6  |
| 2.      | Felhasználói dokumentáció .....                    | 7  |
| 2.1     | Általános ismertető.....                           | 8  |
| 2.2     | Rendszerkövetelmények.....                         | 9  |
| 2.3     | A program telepítése .....                         | 10 |
| 2.4     | A program használatának leírása.....               | 13 |
| 2.4.1   | Játékos szemmel .....                              | 13 |
| 2.4.1.1 | Bejelentkezés.....                                 | 14 |
| 2.4.1.2 | Főoldal.....                                       | 15 |
| 2.4.1.3 | Gazdaság .....                                     | 17 |
| 2.4.1.4 | Építkezés .....                                    | 18 |
| 2.4.1.5 | Kereskedelem .....                                 | 20 |
| 2.4.1.6 | Üzenetek.....                                      | 20 |
| 2.4.1.7 | Exit .....   | 21 |
| 2.4.2   | Adminisztrátori szemmel .....                      | 22 |
| 3.      | Fejlesztői dokumentáció.....                       | 24 |
| 3.1     | Témaválasztás .....                                | 24 |
| 3.2     | Alkalmazott fejlesztői eszközök.....               | 25 |
| 3.3     | Tervezési módszer.....                             | 25 |
| 3.4     | Adatmodell leírása.....                            | 29 |
| 3.4.1   | Adatbázis táblák, mezők.....                       | 29 |
| 3.5     | Részletek feladat-specifikáció, algoritmusok ..... | 30 |
| 3.6     | Tesztelési Dokumentáció .....                      | 38 |
| 3.7     | Továbbfejlesztési lehetőségek .....                | 40 |
| 4.      | Zárószó.....                                       | 42 |
| 5.      | Irodalomjegyzék.....                               | 43 |

## 1. Előszó

Témaválasztásom oka a régmúltban keresendő. Még általános iskolásként kezdtem el különböző online játékokkal játszani.

Körülbelül 14 éves lehettem, mikor először találkoztam a geopolitikai szimulátor fogalmával és ez azonnal elragadott. Már akkor tudtam, hogy egyszer elindítok a saját elképzelésem alapján egy ilyen játékot.

A szoftverfejlesztői tanulmányaim megkezdését főként ez a cél motiválta.

Mostanra az ilyen szimulátor játékok többsége leállt, ám az érdeklődés továbbra is elég nagy, ezáltal nem is volt kérdés számomra, hogy a szakdolgozatom témája egy ilyen jellegű játék fejlesztése lesz.

## 1.1 A téma ismertetése

A Nemzetek egy fórum alapú, körökre osztott szerepjáték, melyet a geopolitikai szimulátorok közé sorolhatunk.

A geopolitikai szimulátor olyan szerepjáték, ami egy fiktív földi politikai közösség alakításáról szól.

A játék ötvözi a klasszikus körökre osztott stratégia (mint például a Total War széria) és a hagyományos interakción alapuló szerepjátékok számtalan tulajdonságát.

Minden játékos egy-egy nemzet uralkodói szerepkörét veszi fel.

A választott ország politikai, gazdasági és minden egyéb ügyeit igazgatja belátása szerint, ügyelve a történelmi kor sajátosságaira.

A játékosok átformálhatják a valós történelmi eseményeket aszerint, ahogy nekik az jól esik.

A játék két fő területen zajlik.

*Program:* gazdasági, stratégiai, kereskedelmi lépések

*Fórum:* szerepjáték, diplomácia

A programban biztosított a titkos megbeszélések lebonyolítása az üzenetküldő rendszernek köszönhetően.

## 1.2 Köszönetnyilvánítás

Rengeteg új szemlélettel, tudással gazdagodtam az iskolában töltött évek alatt. Belepillanthattam milyen kihívások, megoldásra váró feladatok jöhetnek szembe utam során. Igyekszem a kapott tudást felhasználni és élni a lehetőséggel, hogy ebben a nagyszerű szakmában végezhessem mindennapi munkámat.

Meg szeretném köszönni az osztályfőnököm, Szabó Péter támogatását, akihez mindig fordulhattam, ha segítségre volt szükségem.

Különösképpen Édesanyámnak szeretném megköszönni, amiért végig támogatott tanulmányaim során. Bízott, dicsért és a nehéz pillanatokban tartotta bennem a lelket.

## 2. Felhasználói dokumentáció

A Felhasználói dokumentáció pontjai alatt szeretném bemutatni felhasználói szempontból a munkámat.

A leírásoknál egyszerűbb nyelvezettel, az oldal használójának (játékosnak, adminnak) szeretnék segíteni, hogy ne maradjon előtte rejtett, meg nem értett funkció.

## 2.1 Általános ismertető

A Nemzetek egy fórum alapú, körökre osztott szerepjáték, melyet a geopolitikai szimulátorok közé sorolhatunk.

A játék ötvözi a klasszikus körökre osztott stratégia (mint például a Total War széria) és a hagyományos interakción alapuló szerepjátékok számtalan tulajdonságát.

Minden játékos egy-egy nemzet uralkodói szerepkörét veszi fel.

A választott ország politikai, gazdasági és minden egyéb ügyeit igazgatja belátása szerint, ügyelve a történelmi kor sajátosságaira.

A játékosok átformálhatják a valós történelmi eseményeket aszerint, ahogy nekik az jól esik.

A játék megkezdéséhez az alábbiakra van szükség:

- fórumos regisztráció
- nemzet igénylése

A fórumon történő regisztrációval a leendő játékos elfogadja a játékszabályokat (melyeket regisztráció nélkül is megtekinthet a megfelelő topikokban).

Ezt követően pedig a fórum "Nemzet igénylése" nevű topikban kérhet magának országot. A játék üzemeltetője (továbbiakban admin) dönthet arról, hogy a regisztrált megkaphatja az igényelt nemzetet. Ha pozitív elbírálás alá esik a kérelem, a felhasználó megkapja a fórumon a program használatához szükséges belépési adatokat (felhasználónév, jelszó).

A játékprogramhoz tartozó regisztrációt minden esetben az admin végzi, másnak nincs hozzá jogosultsága.



## 2.2 Rendszerkövetelmények

A program futtatásához Windows operációs rendszerre van szükség. A tesztelésre Windows 10 rendszeren került sor, így aztán ezen verziójú operációs rendszer esetében ajánlott a használat.

Mac rendszeren, Linuxon és egyéb más operációs rendszereken nagy valószínűséggel nem fog működni.

Ezenkívül feltétlenül szükséges a .NET Framework 4.7.2 vagy ennél frissebb keretrendszer megléte.

Minimum az 1280\*720-as felbontást támogatja. A program reszponzív, azaz ettől nagyobb felbontásokon is tökéletesen működik, kinagyítható teljes képernyőssé. Ez esetben sem csúszik szét a kép.

A fent említett 1280\*720-as felbontás alatt már nem fér el a jelzón a megnyitott ablak és ez megjelenési hibákhoz vezethet.

Körülbelül 7-8 MB tárhelyet foglal, nagyjából 30-40 MB memóriát fogyaszt működés közben.

## 2.3 A program telepítése

### Telepítés

A program telepítéséhez szükségünk van az alábbi képen látható fájlra:

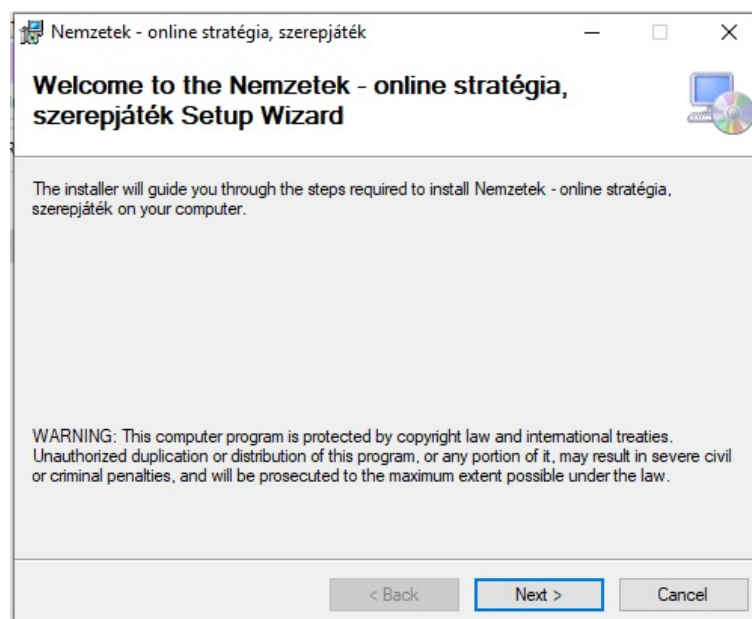
(7. ábra - Telepítő 1)



A képen is látható "Setup" nevű alkalmazásra (teljes nevén: Setup.msi) kattintva indíthatjuk el a telepítőt.

A következő lépésben az alábbi ablak jelenik meg:

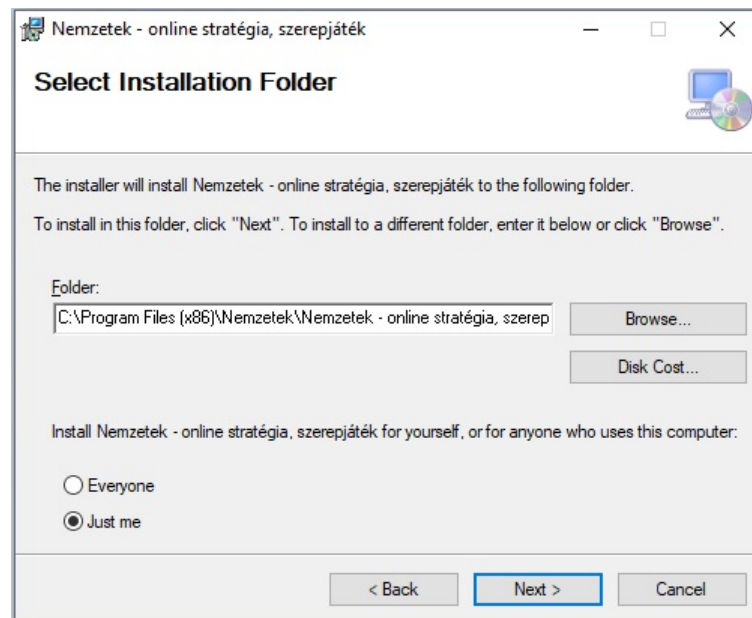
(8. ábra - Telepítő 2)



Ezen a felületen a szerzői jogokról tájékoztat minket a telepítő. Nincs más teendőnk a folytatáshoz, mint a "Next" gombra kattintani.

Megjelenik a következő ablak:

(9. ábra - Telepítő 3)



Itt, ahogy a képen is látható, eldönthetjük, hogy hová szeretnénk telepíteni a programot. Alapértelmezetten a rendszermeghajtó "Program Files (x86)" nevű mappája van beállítva.

Változtatáshoz a "Browse" gombra kattintsunk és navigáljunk el egy számunkra szimpatikus mappáig.

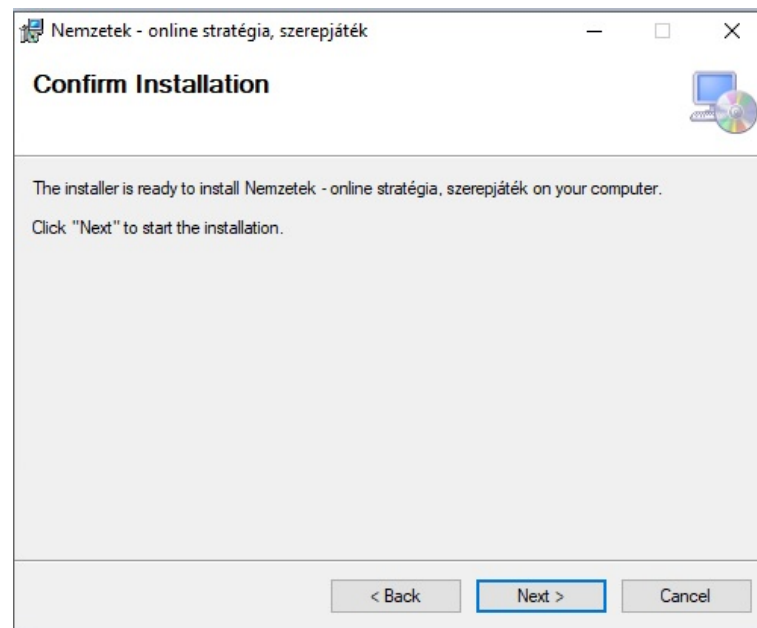
A "Disk Cost"-ra kattintva tájékozódhatunk a számítógépen lévő szabad tárhelyről.

Ezen kívül pedig kiválaszthatjuk azt is, hogy csak az aktuális ("Just me") - ami az alapértelmezett beállítás - vagy pedig minden ("Everyone") felhasználó számára települjön a program.

Ha mindezzel végeztünk, kattintsunk a "Next" gombra.

A telepítő felkészül az installálás folyamatára:

(10. ábra - Telepítő 4)



Kattintsunk a "Next" gombra a folytatáshoz.

Ekkor felugorhat egy figyelmeztetés, ami megerősítést kér a telepítés folytatásához.

Kattintsunk az "Igen"-re.

A telepítés folyamata megkezdődik és pár másodperc alatt be is fejeződik.

A telepítés végeztével a "Close"-ra kattintsunk.

A telepítés során az "Asztalon" megjelenik egy parancsikon "Nemzetek" névvel. Ezzel indíthatjuk el a programot, amikor szükségünk van rá.

### **A program eltávolítása**

A 7. ábrán látható Setup.msi-re kattintva a felugró ablakban eldönthetjük, hogy amennyiben sérült a program valami miatt, javíthatjuk azt. Alapértelmezetten ez van bejelölve.

Ez esetben a telepítő újra lefut és kijavítja az esetleges hibákat.

Törlés esetében a "Remove Nemzetek - online stratégia, szerepjáték"-t jelöljük be és kattintsunk a "Finish" gombra.

Ekkor megjelenhet egy ablak, ami megerősítést vár a folytatáshoz. Kattintsunk az "Igen" gombra.

Elkezdődik az un installálás folyamata, amit egy csík jelez. Amint végbemegy a folyamat, a "Close" gombra kattintsunk.

## 2.4 A program használatának leírása

A leírást két részre bontom. Két felhasználói típus használhatja a programot: a játékos és az adminisztrátor. Ennek megfelelően két különböző felületet találhatunk meg a programban.

### 2.4.1 Játékos szemmel

A játékos felhasználók által elérhető felület, azaz a játéktér, a bejelentkezést követően érhető el. A játékosok ezen a felületen keresztül az adminisztrátori jogosultságokat nem tudják elérni.

Bejelentkezéshez szükséges adatok:

Felhasználónév: teszt2 Jelszó: 123

Felhasználónév: anglia Jelszó: 123

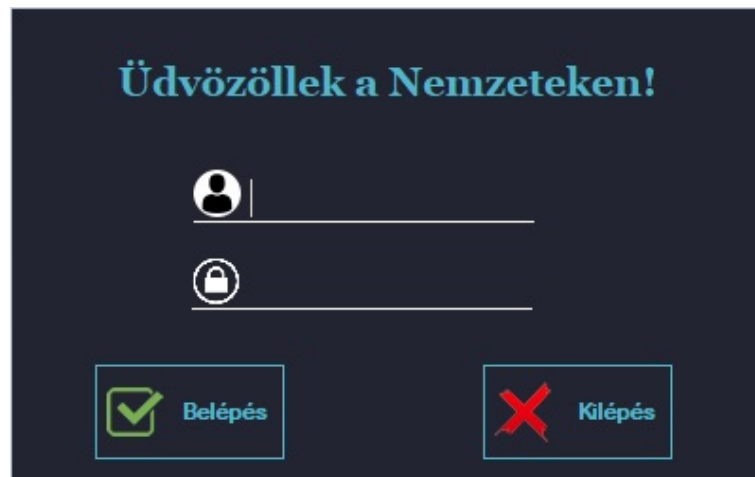
Felhasználónév: magyarkir Jelszó: 123

Felhasználónév: avignon Jelszó: 123

### 2.4.1.1 Bejelentkezés

A program elindulásakor egy bejelentkező felület fogadja a felhasználót.

(1. ábra - Bejelentkező felület)



Itt lehetőségünk van beírni a fórumon az admin által biztosított felhasználónevet és jelszót.

A felhasználónév a felső, a jelszó pedig az alsó sorba írandó. Ezt jelzi az első sor elején található fejecske és a második sor elején lévő lakat.

Miután beírtuk a szükséges adatokat a megfelelő helyre, a "Belépés" nevű gombra kattintva (melyen egy zöld pipa is található) tovább léphetünk a programban.

Ha meggondoltuk magunkat, akkor a "Mégse" (piros X van rajta) gombra kattintva elhagyhatjuk, bezárhatjuk a programot.

## 2.4.1.2 Főoldal

A bejelentkezést követően a Főoldal tárul elénk.

(2. ábra - Főoldal)

**Nemzetek**  
Online stratégia, szerepjáték

AKTUALIS: -  
KÖR: 1.0.0.0 verzió

**Tesztország2**  
Ha résztvevői vagyunk, tragédia, ha csak nézői, komédia. - Aldous Huxley

Nincsen új üzenet

**Általános adatok**

|                             |          |
|-----------------------------|----------|
| Teljes népesség:            | 500 000  |
| Születések száma:           | 59 930   |
| Halálozások száma:          | 12 090   |
| Változás:                   | 47 840   |
| Terület:                    | 7 000    |
| Technológiai kor:           | Középkor |
| Mezőgazdaság teljesítménye: | 5        |

**Gazdasági összefoglaló**

| Nyersanyag | Várható Raktár készlet | Elérhető Források |
|------------|------------------------|-------------------|
| Pénz       | 499 330                | 500 100           |
| Élelmiszer | 1 479                  | 899               |
| Építőanyag | 990                    | 975               |
| Textil     | 1 009                  | 989               |
| Fegyver    | 1 000                  | 1 000             |

**Játéksimertető**

A Nemzetek egy fórum alapú, körökre osztott szerepjáték, melyet a geopolitikai szimulátorok közé is sorolhatnánk. Olyan szerepjáték, ami egy fiktív földi politikai közösség alakításáról szól.

Minden játékos egy-egy nemzet uralkodói szerepét veszi fel. A választott ország politikai, gazdasági és minden egyéb ügyét gazdálja belátása szerint, ügyelve a történelmi kor sajátosságaira. A játékosok átfomálhatják a valós történelmi eseményeket aszerint, ahogy nekik az jól esik.

Időben elhelyezés:  
1400. január elsejével vette kezdetét, tehát a középkorban játszódik, nem sokkal a nagy felfedezések idejét megelőzően, az egyházszakadást követően.

Játékmenet:  
Minden játékos egy (nem a valós adatokkal megegyező) országot irányít.

A választott nemzet vezetőiként kell helytállni: Különböző diplomáciai, politikai, gazdasági döntéseket kell meghozni. A játékos belebújhat egy országot vezető politikai bőrébe, helyette / az ő nevében hozhat meg döntéseket, miközben a többi ország vezetőjével (játékosokkal) kell egyeztetni, vitázni, szövetkezni, stb.

A játékban egy-egy kor egy gazdasági vállalat jelent, azaz bizonyos időnként van váltás (rendszerint hetente). Ekkor érvényesülnek azon döntések, melyek az országra befolyással lehetnek. Ekkor pedig a fórumon (nagykovetéseken, különböző tárgyalási helyszíneken) folyik a diplomácia (egyeztetések, tárgyalások, viták).

Felület:  
- fórum  
- program

A fórumon zajlik az országok diplomáciája, itt tudják a játékosok egymással felvenni a kapcsolatot, tárgyalni, segítségért gyártani és olvasni stb.

A program az országok gazdaságában játszik szerepet. Ezen a

Két részre bonthatjuk a felületet:

Bal oldalon a menürendszer, jobb oldalon pedig az információs panel látható.

A menürendszer navigál el minket a további menüpontok felé.

(3. ábra - Menürendszer)



A jobb oldalon található Információs panel gondoskodik arról, hogy a menüpontoknak megfelelő tartalom betölthessen.

A Főoldalon az országunkkal kapcsolatos legfontosabb információkat láthatjuk és a játék ismertetőjét.

Az alábbiakban, felsorolásszerűen tárgyalom a látható tartalmat:

- az aktuális kör, azaz a játékidő,
- a program verziószáma,
- az ország demográfiai adatai, területe,
- az elért technológiai kor,
- a mezőgazdaság teljesítménye számmal, azaz osztályzata

látható a panel jobb oldalán.

Középen

- az általunk vezetett nemzet neve,
- egy idézet valamely történelmi személytől, könyvből, filmből,
- az érkezett üzenetek száma (ha van ilyen),
- a gazdasági összefoglaló egyszerűsített változata

látható.

A gazdasági összefoglaló részletesebb leírását *az alábbiakban ismertetem a 2.4.1.3 Gazdaság menüpontnál.*

Jobb oldalon pedig a játékismertető látható, mely betekintést ad a játék működésébe.



### 2.4.1.3 Gazdaság

A Gazdaság menüpontra kattintva egy olyan oldalt láthatunk, ahol négy különböző menüpont vár ránk.

#### Gazdasági összefoglaló

Ezen menüpontban látható az ország gazdasági helyzete.

(5. ábra - Gazdasági összefoglaló)

| Gazdasági összefoglaló |                       |         |        |      |        |                        |                   |
|------------------------|-----------------------|---------|--------|------|--------|------------------------|-------------------|
| Nyersanyag             | Induló Raktár készlet | Bevétel | Kiadás | Piac | Profit | Várható Raktár készlet | Elérhető Források |
| Pénz                   | 500 000               | 1 580   | 2 350  | 100  | -670   | 499 330                | 497 750           |
| Élelmiszer             | 1 000                 | 580     | 101    | 0    | 479    | 1 479                  | 899               |
| Építőanyag             | 1 000                 | 15      | 25     | 0    | -10    | 990                    | 975               |
| Textil                 | 1 000                 | 20      | 11     | 0    | 9      | 1 009                  | 989               |
| Fegyver                | 1 000                 | 0       | 0      | 0    | 0      | 1 000                  | 1 000             |

Itt láthatjuk, hogy milyen raktárkészlettel kezdtük az adott kört, mekkora bevételeink, kiadásaink vannak. Ezenkívül a piaci tevékenységeinket is itt tudjuk nyomon követni.

A "Profit" oszlop összegzi a bevételeinket, kiadásainkat.

A "Várható raktárkészlet" oszlopban látható, hogy a következő körben milyen nyersanyagkészlettel indulunk.

Az "Elérhető források" oszlopban a még elkölthető nyersanyagok mennyiségét tudjuk ellenőrizni.

#### Demográfia

Ebben a menüben (ahogy a Főoldalon) áttekinthetőek az ország demográfiai adatai, kiegészülve egy diagrammal.

#### Adózás, monetáris politika

Látható a már beállított adósáv mértéke százalékban és az azzal járó bevétel. Itt állítható az adókulcs, ami a pénzbeli bevételeinket jelenti.

Az adósáv befolyással van a demográfiai adatokra és a lakosság általános moráljára.

A monetáris politika részénél látható az ország pénzemének neve és árfolyama. Lehetőségünk van a pénzmennyiség-céloknál pénzt bevonni a gazdaságba, illetve kivonni a forgalomból. Előbbi esetében a pénzünk árfolyama csökken, utóbbinál viszont növekszik.

Ennek a nemzetközi kereskedelemben van fontos szerepe.

### **Közzolgáltatások**

Itt van lehetőségünk az egészségügy, oktatás és közbiztonság támogatására.

E három szolgáltatás biztosítása az állam feladata, ezek színvonalát az államnak kell biztosítani.

A beállított támogatás pénzben értendő és hatással van a lakosság moráljára, demográfiai adatokra.

A lakosság általános morálja alább tekinthető meg. Fontos magas szinten tartani, hiszen az alacsony morál negatívan befolyásolhatja a játékmenetet.

#### **2.4.1.4 Építkezés**

Az építkezés menüben az ország gazdaságát befolyásoló és a hadsereg számára fontos védelmi épületek találhatók meg.

Jobb oldalt felül látható az aktuális technológiai kor, mely befolyásolja, hogy milyen épületeket használhatunk. Csak a kornak megfelelő épületeket láthatjuk.

Négy menüpont közül választhatunk.

### **Épületek**

Két oszlopot látunk. A meglévő és az építés alatt álló épületek és mennyiségük tekinthető meg.

## **Építkezés**

Az Építkezés menüben a technológiai kornak megfelelő objektumok építését kezdhethetjük meg, vagy vonhatunk meg már egy folyamatban lévő.

A kiválasztott épület árát a két gomb alatt láthatjuk. Minden építkezés 1 körig tart, azaz csak a következőben fejeződik be.

## **Termelési adatok**

A különböző épületek termelési adatai itt érhetőek el. Ez alapján dönthetünk arról, hogy milyen beruházásokat szeretnénk kezdeményezni.

## **Egyéb adatok**

A domborzati adatok adnak tájékoztatást arról, hogy milyen tájegységeket milyen arányban birtokol az országunk.

Háromféle lehetőség van:

- síkság
- dombság
- hegység

Mindegyik befolyással van arra, hogy a termelő épületekből mennyit lehet maximum építeni.

Az építési korlát résznél látható, hogy miből mennyi épületünk lehet összesen. Az egy sorban felsorolt épületekből összesen lehet annyi, amennyit darabot ír a program.

Példa: "Farm, Malom, Legelő" sorban "14 db." van kiírva. Ez azt jelenti, hogy a farmok, malmok és legelők száma összeadva lehet 14 darab.

#### 2.4.1.5 Kereskedelem

A menüpont megnyitásakor a "Piaci ajánlatok" táblázatot pillanthatjuk meg először. Itt válogathatunk a más játékosok által irányított országok piaci ajánlatai között.

Vásárláskor a saját pénzünk árfolyama szerinti árat láthatjuk. Nem kell bajlódni az átváltással. Mindkét fél a saját árfolyama szerint kereskedik.

Vásárolni úgy tudunk, ha a nekünk szimpatikus ajánlat "ID"-jét a "Vásárlás" felirat alatt beállítjuk lásd: a "Vásárlás" feliratú gombra kattintunk. Amennyiben van elég pénzünk a vásárláshoz, a tranzakció azonnal végbemegy. Ez azt jelenti, hogy az általunk kifizetett pénz levonásra kerül (és a hirdető megkapja), a kínált terméket pedig megkapjuk.

A kereskedelemből származó bevételeinket az aktuális körben is felhasználhatjuk.

A saját piaci ajánlatunk megtételéhez ki kell választanunk az "Ajánlattétel" felirat alatt az eladásra kínált nyersanyag típusát. Be kell állítani a mennyiséget és a darabonkénti egységárat. Ha rendelkezünk elegendő elérhető forrással, akkor az ajánlattétel sikeres.

Az ajánlatunk törlésére is lehetőségünk van. Ehhez az "Ajánlat törlése" felirat alatt kell beállítani a saját hirdetésünk "ID"-jét és rányomni a "Törlés" gombra.

#### 2.4.1.6 Üzenetek

A fórumon nem ellenőrizhetőek a privát üzenetek. Ezáltal pedig nem tekinthetőek a fórumos privát üzenetek a játék részének.

Ezt kiküszöbölendő jött létre ez a levelező rendszer.

Amikor megtekintjük ezt a menüpontot, akkor a bejövő üzeneteinket láthatjuk. Ebben a menüpontban intézhetjük a titkos diplomáciai tárgyalásainkat.

### **Bejövő üzenetek**

Ha van új üzenet akkor egy "OLVASATLAN ÜZENET" jelenik meg, feltűnő világos háttérrel.

Az elolvasott üzenetek sötét alapon világos betűkkel láthatóak.

Minden üzenetnél olvasható a dátum, amikor elküldték a levelet, a feladó és az üzenet tárgya.

Ha megnyitjuk az üzenetet, akkor rögtön válaszlevelet is küldhetünk a "Válasz" gombra kattintva. Ekkor megnyílik egy új ablak, ahol a címzett, a tárgy és az eddigi üzenetváltás szövege szerepel. Ezesetben a címzett nem módosítható.

Ha a megnyitott üzenetet be szeretnénk zárni, akkor a jobb felső sarokban lévő "X"-re kattintva léphetünk ki.

### **Elküldött üzenetek**

Itt találhatjuk meg az általunk elküldött üzeneteket. A beérkezettekhez hasonlóan dátummal, (itt már) címmel és a tárggyal.

Ha megtekintjük valamely üzenetünket, akkor a megnyíló ablak bal alsó sarkában láthatjuk, hogy a címzett olvasta-e már, vagy sem.

### **2.4.1.7 Exit**

A kilépés gomb. Mikor rákattintunk, megkérdez minket arról, hogy biztosan ki szeretnénk-e lépni.

Kilépni még a jobb felső sarokban található "X"-re kattintva is tudunk, ekkora nem kapunk megerősítő kérdést.

## 2.4.2 Adminisztrátori szemmel

Az adminisztrátori felület olyan jogosultságokkal bír, melyek az átlag felhasználó számára rejtve maradnak.

Bejelentkezéshez szükséges adatok:

Felhasználónév: teszt Jelszó: 123

Bejelentkezést követően az adminisztrátori felület tárul elénk.

Ezen a felületen van lehetőségünk regisztrálni egy országot és a játékmenet számára elengedhetetlen körváltást lebonyolítani.

### Ország létrehozása, regisztráció

Az ablak alján elhelyezkedő "Ország létrehozás" felirat alatt tudunk regisztrálni egy új nemzetet.

(6. ábra - Ország létrehozása, regisztráció)

| <u>Ország létrehozás</u> |                      |                |                      |  |                      |
|--------------------------|----------------------|----------------|----------------------|--|----------------------|
| Államforma:              | <input type="text"/> | Síkság:        | <input type="text"/> | Ország neve:                           | <input type="text"/> |
| Tech kor:                | <input type="text"/> | Domság:        | <input type="text"/> | Felhasználónév:                        | <input type="text"/> |
| Pénznem neve:            | <input type="text"/> | Hegység:       | <input type="text"/> | Jelszó:                                | <input type="text"/> |
| Pénznem értéke:          | <input type="text"/> | Tengerpart:    | <input type="text"/> | <b>Felhasználói fiók regisztrálása</b> |                      |
| Lakosság méret:          | <input type="text"/> | Napsütés kat.: | <input type="text"/> |  |                      |

Minden nemzetnek 11 különböző tulajdonságot kell megadnunk, ahogy az a fenti képen látható. Ezenkívül pedig a hozzátartozó felhasználónevet és jelszót is csatolnunk kell hozzá, ezzel a két adattal tud a játékos belépni a játékba és birtokba venni a létrehozott országot.

### Tulajdonságok:

Államforma: az ország államformáját kell itt meghatározni.

Tech. kor: "Középkor" - jelenleg csak ez az egy létezik, ezt kell megadni.

Pénznem neve: az ország aktuális pénznemének nevét kell beírni.

Pénznem értéke: alapértelmezetten "100"-as értékű.

Lakosság méret: az ország lakosainak számát kell egész számmal beírni (Például: 1000000).

Síkság, Dombság, Hegység: az ország területi adottságainak megfelelő méretet kell meghatározni egész számmal (Például: 1550).

Tengerpart: "0", vagy "1"-es érték lehet. A "0" szerint van tengerpartja az országnak, az "1"-es pedig azt jelenti, hogy nem rendelkezik tengerparttal az ország.

Napsütés Kategória: 1-5 között változhat. Egyelőre nincs hatása a játékmenetre.

Ország neve: a létrehozandó nemzet nevét kell itt megadni. Ez fog megjelenni a Főoldalon.

Miután megadtunk minden adatot, a "Felhasználói fiók regisztrálása" feliratú gombra kattintva tudjuk a regisztrációt befejezni.

### **Körváltás**

A körváltás fontos része a játékmenetnek. Ez az esemény egy gazdasági ciklust zár le és nyit meg egy újat. A szerepjátékot tekintve új időpontban folytatódik a játék, azaz, ha egy kör egy játék béli évnek felel meg, akkor a következő év kezdődik.

A körváltáshoz arra van szükségünk, hogy a legördülő menüből kiválasszuk azt az országot, melynél a váltást el akarjuk végezni és a "Körváltás" nevű gombra kattintva végbe is megy a folyamat.

A "Körváltás megtörtént" felirat alatt tekinthető meg, hogy mely országok esetében végeztük el a műveletet. A program bezárásakor, vagy ha innen a tesztországba lépünk, az itt tárolt adatok elvesznek.

A körváltás azért zajlik egyesével, hogy az inaktívabb játékosoknak is lehetősége legyen felzárkózni.

## **Tesztország gomb**

A jobb felső sarokban található egy "Tesztország" feliratú gomb. Erre kattintva beléphetünk a játékosok által látható felületre. Ezzel a teszt nemzettel lehetőségünk van kipróbálni a funkciókat.

Vissza is léphetünk a Főoldalról az adminisztrátori felületre a "Verziószám" és az "Általános adatok" között található gombra kattintva. Ezt a gombot csak az adminisztrátor láthatja.

## **3. Fejlesztői dokumentáció**

### **3.1 Témaválasztás**

Régóta szerettem volna a Nemzetek-et létrehozni, megvalósítani. Az iskolai tanulmányaim elindítottak azon az úton, amin elsajátíthatom azt a tudást, ami szükségeltetik a játék elkészítéséhez.

Azért választottam a C# programozási nyelvet a szakdolgozatom témájához, mert úgy érzem, hogy kellő magabiztossággal mozgok ebben a technológiában. Úgy ítélt meg, hogy a játék minden funkcióját képes vagyok megvalósítani ezzel a nyelvvel.

Azt gondolom, hogy a szakdolgozatom által megvalósíthatom egy régi álmom és közben remek referenciát is biztosíthat számomra, hogy szoftverfejlesztőként érvényesülni tudjak a munkaerőpiacon.



## 3.2 Alkalmazott fejlesztői eszközök

Felsorolásszerűen bemutatom a fejlesztés során használt programokat, programozási nyelvet, adatbáziskezelő rendszert, fejlesztői környezetet, modulokat.

Alkalmazott programozási nyelv:

- C#

Alkalmazott keretrendszerek, modulok:

- .NET Framework 4.2.7
- MetroModernUI 1.4.0 (Visual Studio NuGet-en keresztül letölthető)

Alkalmazott szoftverek:

- Visual Studio (programozáshoz)
- Corel® PaintShop® Pro 2020 (ikonok készítéséhez)
- Microsoft Word (dokumentáció készítéshez)

Adatbázis-kezelő szoftver és technológia:

- MySQL
- phpMyAdmin

## 3.3 Tervezési módszer

A tervezés során az átláthatóságot, kezelhetőséget tartottam szem előtt. Fontosnak tartottam, hogy a logikailag különböző, más feladatot ellátó kódrészleteket egymástól jól látható módon elkülönítsem.

Törekedtem arra, hogy a továbbfejlesztés minél könnyebben megvalósítható legyen, mikor egy-egy funkció megváltoztatására sor kerül, vagy újabbakat adok a programhoz.

Ennek érdekében tanulmányoztam az MVC szemléletet és az OOP lehetőségeit.

## **OOP**

Objektumok használatával egyszer szerkezet, jól kézben tartható programok készíthetők. Az objektumorientált programozás középpontjában az egymással kapcsolatban álló programegységek hierarchiájának megtervezése áll.

Az adatokat és az adatokon végrehajtható műveleteket egyenrangúan, zárt egységben kezeljük. Ezeket az egységeket objektumoknak nevezzük, mely nem más, mint az adatok és az adatokat kezel alprogramok (metódusok) egységbezárása.

Az objektumorientáltság három fő ismérve:

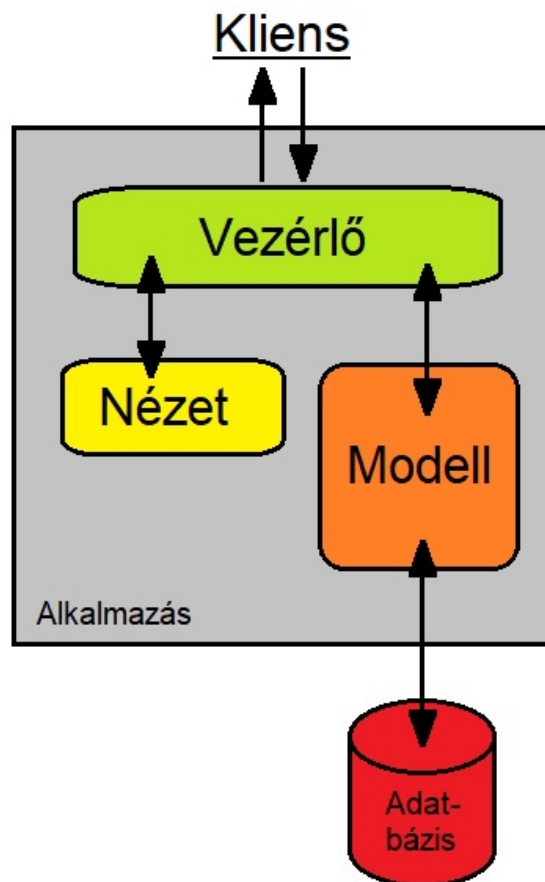
- Egységbe zárás: mikor egy egységként kezeljük, egy osztályban tároljuk logikailag összetartozó adatstruktúrákat, metódusokat.
- Öröklődés: Azt jelenti, hogy az adott meglévő osztályokból levezetett újabb osztályok öröklík a definiálásukhoz használt alaposztályok már létező adatstruktúráit és metódusait
- Többértékűség: Azt értjük ezalatt, hogy egy örökölt metódus az adott objektumpéldányban felüldefiniálódik.

Munkám során legfőképpen az egysége zárás elvét igyekeztem követni.

## MVC

A modell-nézet-vezérlő (MNV) (angolul model-view-controller) a szoftvertervezésben használatos programtervezési minta.

(11. ábra - MVC tervezési minta)



Összetett, sok adatot a felhasználó elé táró számítógépes alkalmazásokban gyakori fejlesztői kíváncsi az adathoz (modell) és a felhasználói felülethez (nézet) tartozó dolgok szétválasztása, hogy a felhasználói felület ne befolyásolja az adatkezelést, és az adatok átszervezhetők legyenek a felhasználói felület változtatása nélkül. A modell-nézet-vezérlő ezt úgy éri el, hogy elkülöníti az adatok elérését és az üzleti logikát az adatok megjelenítésétől és a felhasználói interakciótól egy közbülső összetevő, a vezérlő bevezetésével.

### Modell:

Az alkalmazás által kezelt információk tartomány-specifikus ábrázolása. A tartománylogika jelentést ad a puszta adatnak (pl. kiszámolja, hogy a mai nap a felhasználó születésnapja-e, vagy az összeget, adókat és szállítási költségeket egy vásárlói kosár elemeihez).

Sok alkalmazás használ állandó tároló eljárásokat (mint mondjuk egy adatbázis) adatok tárolásához. Az MNV nem említi külön az adatelérési réteget, mert ezt beleérti a modellbe.

### Nézet:

Megjeleníti a modellt egy megfelelő alakban, mely alkalmas a felhasználói interakcióra, jellemzően egy felhasználói felületi elem képében. Különböző célokra különböző nézetek létezhetnek ugyanahhoz a modellhez.

### Vezérlő:

Az eseményeket, jellemzően felhasználói műveleteket dolgozza fel és válaszol rájuk, illetve a modellben történő változásokat is kiválthat.

Fontosnak tartottam, hogy a különböző programrészeket jól látható módon elkülönítve tároljam. Ezáltal sokkal könnyebbé vált a program tervezése, fejlesztése egyaránt.

Megoldásomban az adatok beolvasását a "Modell" végzi, melynek adatait a "Nézet" réteg jelenít meg a felhasználó számára. E kettő között a "Vezérlő" teremti meg a kapcsolatot a különböző számítások elvégzésével.

## 3.4 Adatmodell leírása

Az adatbázis létrehozásakor szöveges kódolásnak az "utf8mb4\_hungarian\_ci"-t választottam.

Ez az illesztés hibátlanul kezeli a magyar abc karaktereit és nem különböztet meg kis és nagybetűk között különbséget.

### 3.4.1 Adatbázis táblák, mezők

Törekedtem arra, hogy a program menüpontjainak megfelelően külön-külön táblákat hozzak létre.

#### **ID mezők**

Minden táblának az első mezője az azonosítója, amivel a munka során

hivatkozhatok az értékekre. Ezek az ID mezők a felhasználók előtt rejtve maradnak többnyire. Az ID mezők int típusú változóba vannak fel véve, hiszen csak számot tartalmaznak.

AutoIncrement beállítást használtam így az értéket automatikusan határozza meg eggyel növelve az előző rekord értékéhez képest, így két ugyanolyan azonosító soha nem lesz egy táblán belül.

Ez a mező a legfontosabb tartalmunk a táblák összekapcsolásánál. Az ID (kivéve a Marketplace és buildings, messages tábla esetében) a felhasználó játékos adatait kapcsolja össze.

A programban a belépést követően mentésre kerül a userID és ezt felhasználva történik a kapcsolatteremtés a többi oszloppal.

## "users" tábla

Itt tárolódnak el a regisztrációnk legfontosabb elemei:

- ID
- felhasználónév
- jelszó
- jogosultság

## 3.5 Részletek feladat-specifikáció, algoritmusok

Ebben a fejezetben az általam fontosnak vélt metódusokkal foglalkozom bővebben.

1.

A C\_Login.cs fájlban található kód az sql injekciót hivatott megelőzni.

*(12. ábra - Paraméteres átadás)*

```
MySQLParameter param = new MySQLParameter();  
param.ParameterName = "@login_name";  
param.Value = username;  
MySQLCommand cmd_username = new MySQLCommand(sql_name, Model_db.conn);  
cmd_username.Parameters.Add(param);
```

Hosszas utánajárást követően arra jutottam, hogy a paraméteres átadás megoldása lehet kellően hatékony eszköze annak, hogy az sql injekciót megelőzzem. Fontosnak tartom, hogy illetéktelenek ne tudjanak kárt tenni az adatbázisban, vagy admin rangot szerezni.

A paraméter nevének beállítom a "@login\_name", illetve "@login\_password" neveket, melyek a megfelelő sql lekérdezésbe kerülnek. A létrehozott paraméternek átadom a LoginF.cs-ben tárolt felhasználónevet és jelszót tároló textbox értékét. Ezt követően pedig egy MySQL commandnak átadom a paramétert.

2.

A C\_Login.cs fájlban található Encrypt() metódus a titkosítást hivatott elvégezni.

Az Encrypt() metódus visszatérési értékének hash kódja kerül összehasonlításra az adatbázisban szereplő adatokkal.

*(13. ábra - Titkosítás 1)*

```
string Encrypt(string value)
{
    using (MD5CryptoServiceProvider md5 = new MD5CryptoServiceProvider())
    {
        UTF8Encoding utf8 = new UTF8Encoding();
        byte[] data = md5.ComputeHash(utf8.GetBytes(value));
        return Convert.ToBase64String(data);
    }
}
```

A felhasználó beírja a jelszavát, ez a string átadásra kerül az Encrypt() módszernek, mely egy hosszú stringet ad visszatérési értéként.

*(14. ábra - Titkosítás 2)*

```
password = Encrypt(tb_password).GetHashCode().ToString();
```

---

Ennek a stringnek a hash kódját használom fel.

3.

A Model\_db.cs fájlban található „list\_upload\_vertical()” nevű módszer az adatbázisban lévő adatok oszlop szerinti beolvasását végzi.

*(15. ábra - adatbeolvasás vertikálisan)*

```
public void list_upload_vertical(string sql_query, List<string> list, string db_column)
{
    cmd.Connection = conn;
    cmd.CommandText = sql_query;
    try
    {
        conn.Open();
        reader = cmd.ExecuteReader();
        while (reader.Read()) { list.Add(reader.GetString(db_column)); }
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
    finally { conn.Close(); }
}
```



A metódus három dolgot vár:

- egy sql lekérdezést, azaz amit keresünk az adatbázisban
- egy listát, mely tárolja a beolvasott adatokat
- egy stringet, mely az érintett oszlopot kéri, amelynek az adataira kíváncsiak vagyunk.

Ez a metódus akkor hasznos, ha az adatbázis egy táblájában egy oszlop összes adatát szeretnénk megtudni.

Mivel nem tudhatjuk előre, hogy az oszlop hány rekordot tartalmaz, listába olvassuk be az adatokat.

4.

A Model\_db.cs fájlban található list\_upload\_horizontal() nevű metódus az adatbázisban lévő adatok soronkénti beolvasását végzi.

*(16. ábra - adatbeolvasás horizontálisan)*

```
public void list_upload_horizontal(List<string> list_upload_data, string sql_query, List<string> list_sql_columns)
{
    cmd.Connection = conn;
    cmd.CommandText = sql_query;
    try
    {
        conn.Open();
        reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            for (int i = 0; i < list_sql_columns.Count; i++)
            {
                list_upload_data.Add(reader.GetString(list_sql_columns[i]));
            }
        }
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
    finally { conn.Close(); }
}
```

A metódus három dolgot vár:

- egy listát, melybe a beolvasott adatok mentésre kerülnek
- egy sql lekérdezést, azaz amit keresünk az adatbázisban
- egy listát, mely tartalmazza azon tábla oszlopok neveit, melyek rekordjait szeretnénk megtudni.

Ez a metódus akkor hasznos, mikor az adatbázis egy táblájának egyetlen sorának adatait szeretnénk megtudni.

A "list\_sql\_columns" lista azért hasznos, mert ezzel szelektálható, hogy az adott sor mely oszlopainak rekordjait mentsük le.

Az alábbi képen látható példa jól szemlélteti ezt:

*(17. ábra - a horizontális adatbeolvasáshoz tartozó oszlopok listája - példa)*

```
public List<string> list_budget_sql_query_columns = new List<string>()
{
    "storage_money", "storage_food", "storage_building_material", "storage_textile", "storage_weapon",
    "cost_money", "cost_food", "cost_building_material", "cost_textile", "cost_weapon",
    "market_money", "market_food", "market_building_material", "market_textile", "market_weapon", "tax", "health_cost",
    "education_cost", "public_safety_cost"
};
```

A lista elemein végigmegy a for ciklus, így beolvassa a szükséges adatokat.

A "list\_upload\_vertical()" és "list\_upload\_horizontal()" metódusok abban nyújtanak segítséget, hogy a változó mennyiségű adatok beolvasását könnyen, gyorsan meg lehet velük valósítani.

Míg előbbi a táblában függőlegesen, utóbbi a vízszintesen elhelyezkedő adatokat képes beolvasni - azok mennyiségétől függetlenül - egy-egy listába.

5.

A M\_db\_economyf.cs fájlban található "data\_upload()" nevű metódus az adatok frissítéséért felel.

(18. ábra - adatfrissítés 1)

```
public void data_upload(string sql_table, string sql_column, string amount)
{
    cmd.Connection = Model_db.conn;
    cmd.CommandText = $"UPDATE {sql_table} SET {sql_column} = {amount} WHERE ID = {C_Login.ID}";
    try
    {
        Model_db.conn.Open();
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex) { MessageBox.Show("Nem sikerült végrehajtani a feladatot!\nOka: " + ex.Message, "Hibaüzenet",
        MessageBoxButtons.OK, MessageBoxIcon.Error); }
    finally { Model_db.conn.Close(); }
}
```

Három dolgot kér a metódus:

- egy tábla nevet
- egy oszlop nevet
- egy mennyiséget

Ez a metódus a bekért mennyiség alapján kicseréli a megadott tábla oszlopának értékét az adott felhasználóhoz tartozó sorban.

6.

A Model\_db.cs fájlban található "database\_update()" nevű metódus az adatok frissítéséért felel, de picit másként, mint az ötös pontban tárgyalt metódus.

(19. ábra - adatfrissítés 2)

```
public void database_update(string sql_table, string sql_column, string amount)
{
    cmd.Connection = conn;
    cmd.CommandText = $"UPDATE {sql_table} SET {sql_column} = {sql_column} + {amount} WHERE ID = {C_Login.ID}";
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex) { MessageBox.Show("Nem sikerült végrehajtani a feladatot!\nOk: " + ex.Message, "Hibaüzenet",
        MessageBoxButtons.OK, MessageBoxIcon.Error); }
    finally { conn.Close(); }
}
```

Három dolgot kér a metódus:

- egy tábla nevet
- egy oszlop nevet
- egy mennyiséget

Ez a metódus nem kicseréli az adatot, hanem az adatbázis rekordjával egy összeadás műveletet hajt végre.

Példa:

A "data\_upload()" metódus segítségével lehet adót állítani. Ilyen esetben az új értéket kell a régi helyére tenni.

A "database\_update()" metódus segítségével kiszámolható például, hogy ha a Monetáris politikánk szerint pénzt vonunk el a gazdaságból (levonjuk az értéket a meglévőből,) akkor mennyi marad.

Ebben az említett esetben egy negatív számot adunk az adatbázisban szereplő "storage\_money" rekordhoz a "budget" táblában. Ezáltal megkapjuk, hogy mennyivel kevesebb pénzzel rendelkezik az ország.

7.

A DiplomacyForm.cs fájlban a beérkezett, illetve küldött üzenet kezelése történik. Ennek megoldására dinamikus buttont alkalmaztam, melynek a „message\_button\_style()” metódus adja meg a paramétereit.

A dinamikus buttonok egyedi nevet kapnak:

A név első fele jelzi, hogy érkezett (incoming), vagy küldött üzenet (sent). Ezt követi egy "\_" jel, majd pedig egy sorszám kerül a végére. Ez a sorszám az indexe az üzenetnek.

Annak érdekében, hogy a felhasználó mindig a megfelelő üzenetet jeleníthesse meg kattintáskor, a "message\_click()" metódus felel. Ebben a metódusban a "Button btn = sender as Button" kód által meg tudom vizsgálni a kattintott button nevét.

Ehhez a nevet szétválasztom és kiderül, hogy ez érkezett, vagy küldött. Ha az ország nevével egyező a feladó ország neve, akkor küldött üzenetként jelenik meg, ellenkező esetben érkezettként. A button nevében elrejtett szám, az indexe garantálja azt, hogy a megfelelő sorszámú levél nyílik meg.

## 3.6 Tesztelési Dokumentáció

### A tesztelés lehetséges céljai

- Bizonyosságot szerezni a rendszer minőségi állapotáról
- Információ a döntéshozáshoz
- Hibák keresése
- Hibák megakadályozása

Tesztelésre azért van szükség, hogy a szoftver termékben meglévő hibákat még az üzembe helyezés előtt megtaláljuk, ezzel növeljük a termék minőségét, megbízhatóságát. A program főleg azon funkcióit kell tesztelni, amiket a felhasználók legtöbbször fognak használni.

### A tesztelés szintjei

- komponenteszt: csak a rendszer egy komponensét teszteli önmagában forráskód ismeretében.
- integrációs teszt: kettő vagy több komponens együttműködési tesztje.
- rendszerteszt: az egész rendszert, tehát minden komponenst együtt, teszteli.
- átvételi teszt: a felhasználók a kész rendszert tesztelik.

## Tesztelési technikák

A tesztelési technikákat csoportosíthatjuk a szerint, hogy a teszteseteket milyen információ alapján állítjuk elő.

- Fekete-dobozos (black-box) vagy specifikáció alapú, amikor a specifikáció alapján készülnek a tesztesetek.

Amikor a tesztelő nem látja a forráskódot, de a specifikációkat igen, fehérdobozos tesztelésről, amikor a forráskód rendelkezésre áll. A fekete-dobozos tesztelést specifikáció alapúnak is nevezzük, mert a specifikáció alapján készül. Ugyanakkor a teszt futtatásához szükség van a lefordított szoftverre. Leggyakoribb formája, hogy egy adott bemenetre tudjuk, milyen kimenetet kellene adni a programnak. Lefuttatjuk a programot a bemenetre és összehasonlítjuk a kapott kimenetet az elvárttal.

- Fehér-dobozos (white-box) vagy strukturális teszt, amikor a forráskód alapján készülnek a tesztesetek.

Mindig egy már kész struktúrát, pl. program kódot, tesztelünk. A strukturális teszt esetén értelmezhető a (struktúra) lefedettség. A lefedettség azt mutatja meg, hogy a struktúra hány százalékát tudjuk tesztelni a meglévő tesztesetekkel. Általában ezeket a struktúrákat teszteljük: kódsorok, elágazások, metódusok, osztályok, funkciók, modulok. Például a gyakran használt unit-teszt a metódusok struktúra tesztje.

### 3.7 Továbbfejlesztési lehetőségek

Az a tervem, hogy ősszel publikálásra kerül a játék. Ahhoz, hogy ez megtörténhessen, rengeteg funkció beépítésére van még szükség.

#### **Haderő**

A hadsereg elmaradhatatlan kelléke egy olyan játéknak, ahol egy országot kell vezetni. Többféle egységet, azok árát, toborzási rendszerét, tapasztalati rendszerét ki kell dolgozni.

#### **Kereskedelem**

A kereskedelem ki fog bővülni egyedi ajánlatok küldésének lehetőségével. Ez lehet egy irányú vagy két irányú. Egyirányú, ha csak küldünk nyersanyagot és nem kérünk cserébe semmit és két irányú, ha cserébe kérünk valamit.

#### **Sport**

Eddigi észrevételeim szerint a leendő játékosoknak fontos, hogy pár népszerűbb sportág nemzeti válogatottja szerepeltesse magát a játékban. Ez azért lehet érdekes, mert így lehetőség nyílik majd világversenyek tartására, ami tovább erősíti a valószerű szerepjátékot.

#### **Fejlesztések**

Rengeteg tudományos innováció válik elérhetővé a jövőben a játékosok számára. Ezáltal a középkort elhagyhatják és olyan technológiákat érhetnek el később, amit ma (a való életben) is használunk.



## **Diplomáciai táblázat**

Ez egy olyan felület lesz, ahol a játékosok beállíthatják/megtekinthetik, hogy melyik országnak milyen viszonya van a másikkal. Ez a felület segítséget fog nyújtani a diplomáciai kapcsolatok áttekintésében.

## **Törvénykezés**

Egy ország vezetőjének mi lehetne a legfontosabb feladata, mint a törvényalkotás. Hamarosan ez a funkció is elérhetővé válik.

Ezenkívül rengeteg további ötletem van, amit a programba szeretnék beépíteni, de most csak a legfontosabbak kerültek felsorolásra.

Az a tervem, hogy a felhasználók minél korhűbbnek, valóságosabbnak érezzék a játékot.

Az új funkciók beépítése mellett szeretném a kódot még átláthatóbbá, karbantarthatóbbá tenni.

Például:

Az Építkezés menü Termelési adatok almenüjében megjelenő adatok kiírásával nem vagyok elégedett. Jelenleg fix labellel van megoldva, ám ha új termelő épület kerülne be a játékba, vagy pedig kiveszek egyet, akkor ez problémát okozna. Szeretném dinamikus labellel megoldani, ahogy a levelezésnél tettem.

A levelezés menüben a későbbiekben a jobb karbantarthatóság és átláthatóság miatt szeretném a listák helyett egy struktúrában tárolni az érkezett és küldött üzenetek adatait.

## 4. Zárószó

Szakedolgozatom készítése közben az addig megszerzett tudást a gyakorlatba átültetve rengeteg tapasztalatot gyűjtöttem.

Több olyan módszert sikerült kidolgoznom a munkafolyamat során, amivel a későbbi fejlesztéseket nagyban megkönnyítettem.

Belekóstoltam az MVC modell és az OOP használatába, melyek segítségével sokkal átláthatóbb, tisztább kódolást tudtam elérni. A tervem az, hogy a további munkáim során ezen modell és elvek használatával még hatékonyabb, átláthatóbb kódokat tudjak írni.

A MySQL-lel kapcsolatos tudásomat sikerült magasabb szintre emelni, amit nagyon fontosnak tartok, mivel az adatbázisok fontos szerepet játszanak a szoftverfejlesztés területén.

## 5. Irodalomjegyzék

1. Metro Modern UI 1.4.0 - <https://github.com/dennismagno/metroframework-modern-ui>
2. 11. ábra - <https://hu.wikipedia.org/>
3. Microsoft Visual Studio Installer Projects - <https://marketplace.visualstudio.com/>
4. phpMyAdmin - <https://www.phpmyadmin.net/>
5. Titkosítási eljárás (System Security Cryptography Namespace) -  
<https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography?view=net-5.0>