

---

## Game development through independent student activities

R. Berci

*Capsys Kft. Váci út 80-84, Budapest 1133, Hungary, robert.berci@capsys.hu*

---

### Abstract

Nowadays, non-formal (extra-curricular) and informal (non-organized) forms of learning are increasingly appearing. The game development to be presented in the article was also implemented in such an informal environment. The possibility of personal growth in this methodology is much more in programming subjects. The project was implemented online so that smaller country clubs can easily take advantage of the application. The game development steps and implementation details have also been developed, which can provide useful information for other similar students to develop their own tasks.

*Keywords:* extra-curricular; education; software development

---

## Játékfejlesztés önálló hallgatói tevékenység során

Bérci R.

*Capsys Kft. Váci út 80-84, Budapest 1133, Hungary, robert.berci@capsys.hu*

---

### Absztrakt

Napjainkban egyre inkább megjelenik a nem formális, azaz a tanórán kívüli, valamint az informális, azaz a nem szervezett, önálló tevékenység útján megvalósuló tanulási formák. A cikkben bemutatandó játékfejlesztés is egy ilyen nem formális környezetben született. Az ebben rejlő fejlődés lehetősége még inkább igaz a programozási tantárgyak körére. A projekt megvalósítása online felületen történt annak érdekében, hogy a kisebb vidéki klubok is könnyedén élhessenek az alkalmazás előnyeivel. A játékfejlesztés lépései, a megvalósítás részletei is kidolgozásra kerültek, melyek hasznos információkat adhatnak más hasonló hallgatói önálló feladatok kidolgozásához.

*Kulcsszavak:* tanórán kívüli tanulás; oktatás; szoftverfejlesztés

---

### 1. Bevezető

Az egyetemi oktatásban a tanórán kívüli szakmai tudásunk fejlesztése leginkább az önképzés során valósulhat meg. Különösen igaz ez olyan tantárgyak esetében, amelyek nagy projekteken átívelő, gondolkodtató, akár több emberrel való együttműködést kívánó feladatok megoldását kívánja meg. Az ilyen tantárgyak tekintetében a legideálisabb az lehetne, ha az iskola tanulási szférájában az extrakurrikulum egyenrangúként jelenne meg, mivel így a tanulást elősegítő motivációk felerősödhetnek és a differenciálás is széleskörűen megvalósulhatna. (Judit P., 2014), (Buckley, Lee, 2018)

A cikkben a GamesWorkshop által kiadott Warhammer 40 000 elnevezésű játékhoz egy úgynevezett sereg összeállítását segítő alkalmazás fejlesztését mutatja be, mely hallgatói

önálló feladat keretében készült. A játék esetében egy tradicionálisan szabálykönyv, az úgynevezett kódex segítségével papíron, számológéppel lehet összeírni a sereg felépítését. Ez igen hosszadalmas feladatot jelent, az esetleges módosítások nehezen megvalósíthatók. Az összetettséget az jelenti, hogy a játékos miután sereget (frakciót) választott, a pontértékből hősöket, osztagokat, járműveket vehet. Ezeket felruházhatja eltérő felszereléssel, fegyverekkel vagy akár a létszámát is változtathatja további pontok elköltésével.

Warhammer 40 000 elnevezésű játék egy 6' x 4' (183\*122 cm) méretű asztalon zajlik, amin a figurák csatáznak a tereptárgyakkal előre berendezett hadszíntéren, 2-5 órán keresztül. A gyakorlati tapasztalat alapján ezeket a körülményeket nem szeretik otthon megteremteni sem a játékosok, sem családtagjaik. A gyártó ezért a saját boltjaiban is biztosít 2-3 asztalt, mennyiségi és változatos tereptárggyal saját vásárlóinak játék céljára. Kialakultak olyan független klubok is, ahol vásárolni nem is lehetséges, viszont nagyobb területen több asztal és több tereptárgy elérhető. Jelenleg Magyarországon nincs a gyártó által üzemeltetett bolt, így csak a független klubokba járnak le az emberek játszani. A baráti játékok mellett az ilyen kluboknak a fő előnye, hogy nagyobb rendezvényeket tudnak szervezni több résztvevő számára, mint egy bolt. (Carter et al, 2015)

Az ilyen szervezett versenyekre általánosságban igaz, hogy a kiírásukban meg vannak határozva a jelentkezési feltételek. Ilyenek a határidők, a figurák állapota (teljes festett sereg, nem használhatóak proxy figurák), pontérték és esetleges eltérések az eredeti szabályrendszerhez képest. Előfordulhat, hogy a szervezők kizárnak egyes modelleket, ezzel próbálva egyensúlyozni a gyártó által bent felejtett erőviszony eltéréseket, vagy narratív vonalat vinni a versenybe. Mivel azonban a játékon belül már 6 faj és 29 eltérő frakció elérhető, a szervezők sem feltétlen ismerik az összes seregépítési opciót. Egy-egy frakció kódexe 160 oldal is lehet, és elég összetett ahhoz, hogy tévedhessen a nevező játékos, vagy akár direktbe elcsaljon papíron egy két megkötést, ami már számottevő előnyt nyújthat neki az ellenfeleivel szemben.

A versenyszférától jelenleg még messze maradók között is vannak, akik csak alkalmanként játszanak esetleg egymással, vagy tapasztaltabbakkal. Akik ritkábban játszanak szoktak abba botlani, hogy mindenki, akivel játszanak máshogy ismeri ugyanazt a szabályt. Sajnos a magyar tapasztalat, hogy mivel nem anyanyelvi szinten értjük a szabályokat, ezért eltérően is értelmezzük azokat. A szabályok gyakran még a nemzetközi, akár anyanyelvi angolos közösségekben is vitatott a megfogalmazás miatt. A gyártó ezért módosítja, pontosítja a szabályokat, de annyira összetettre nőtte ki magát a rendszer az évtizedek alatt, hogy ez

jelenleg még csak szélmalomharc. Az interneten FAQ-k (frequently asked question - gyakran ismételt kérdések) igyekeznek segíteni a játékosokat, de a játék közben, egy esetleges nézeteltérés miatt internetet bogarászni órákig nem a legjobb játéktapasztalat. A gyártó saját nyilatkozatai alapján, egy figurákat forgalmazó cég és az, hogy játszani is lehet velük, egy másodlagos haszon, ezért a problémák megoldására nem is törekszik. Így születnek a házi szabályok a klubokban és versenyeken, amiket fentebb már említettem.

A játék angolszász eredetű, ezért nem SI mértékegységeken alapul, illetve nincsenek hivatalos, magyar nyelvű szabálykönyvek, kódexek. Népszerűbb nyelvekre, mint a francia vagy német, ugyan készültek gyártói fordítások, amik ki vannak egészítve azzal, hogy kérdéses esetben az eredeti angol leírást kell alapul venni. Ilyen magyar fordításról nincs terv a gyártónál. A magyar közösségnek nagy segítséget nyújtana egy egységes magyar nyelvű szabályrendszer. Vannak a közösségnek oszlopos tagjai, akik gyakorlati tapasztalatukkal ki tudják silabizálni, hogy vajon egy rosszul megfogalmazott angol szabály mögött mi is lehetett a gyártó szándéka. Az egyetértés sajnálatos módon nem minden esetben van meg, és így továbbra is házi szabályokat gyártanak még ha esetleg a ház nagyobb is. A gyakorlati gond azzal van, hogy ezek a házi szabályok nem terjednek, és aki nem ismeri őket, csak a versenyen már későn szembesül vele.

A fentiek orvoslásához szeretnék hozzájárulni az online sereg készítővel. Hiszem, hogy ha létezne egy adatbázis, ami tartalmazza mind a gyártó szabályait eredeti nyelven és magyarul is, akkor a klubok által támogatott házi szabályokat is könnyebb lenne terjeszteni. Feltehetik a saját szabályaikat, és onnantól kereshető minden eltérés. Ezeket az eltéréseket a versenyek kiírásában feltüntetve, a játékosok már fel tudnak készülni rá a meglepetések helyett.

A seregekészítő program, a papíros seregépítést is felválthatja idővel a közösség minden tagjánál. A csalások és tévesztések kizárása mellett lehetővé teszi a játékosok számára, hogy könnyen a versenykiírás korlátozásait is beépíthessék a seregükbe.

A jövőben az ilyen jellegű játékok esetében a modernebb ember-számítógép kapcsolatok is megjelenhetnek, támogatva ezzel a játékélmény fokozását. (Katona et al, 2016) (K J, 2015),

## 2. Fejlesztői és futtatási környezet

A fejlesztéshez a WAMP SERVER csomagot használtam. Ezáltal egy csomagban telepíthettem a MySQL 5.7.14-es verzióját az adatbázishoz és Apache 2.4.23 + PHP 5.6.25

kombinációt a felhasználó felülethez. A szervercsomag mellett szólt, hogy ingyenes és nagy közösség használja. (Meloni et al, 2003), (Zandstra et al, 2005).

A MySQL tökéletes a célra, hiszen ingyenes és rengeteg dokumentáció található hozzá. Ehhez a projecthez kifejezetten bónusz volt, hogy rengeteg ingyenes webhosting is ezt tette elérhetővé. A kezeléséhez a phpMyAdmin-t a telepített csomag részét képezte. (Ullman et al, 1998)

A felhasználói felülethez php-t választottam, mert tudtam, hogy minden esetlegesen előforduló feladatot el tud látni. A felhasználóknak nem kell semmit telepíteni, ezzel is elősegítve a könnyebb csatlakozást.

### 3. Követelmény felmérés

Magát a projekt ötletét az egyik versenyen, a fordulók közötti beszélgetés során találtuk ki. Miközben latolgattuk, hogy kinek ki lesz a következő ellenfele, szembesültünk vele, hogy olyan embert fogok kapni ellenfélnek, akinek a seregéről konkrétan nem tudok semmit. Mivel nekünk nem állt rendelkezésünkre az a kódex, amiben a leendő ellenfelem frakciója szerepel, ezért ezt nem lehetett figyelembe venni a fejlesztés, tervezés során. Azonban a felhasználók részéről felmerült az igény egy olyan rendszer kidolgozására, mely a sereglisztákat szabályosan megépítéséhez ad keretet. Ehhez kell készíteni egy rendszert, amiben le lehet modellezni a szabálykönyvek és kódexek tartalmát, hogy abból lehessen építkezni a sereg felállítása során, és mindez legyen több helyszínen és könnyen elérhető, ugyanazzal a bázis információ halmazzal.

#### 3.1. Működési paraméterek

Ismervén mind a két rendszert (a játékét és a fejleszteni valót), arra jutottam, hogy mivel kluboknak és versenyszervezőknek készül, sem elkülönített felhasználók, sem autentikáció nem fog kelleni. Amennyiben valaki használni kívánja a rendszert, azonosítás nélkül megteheti.

Az adatok kampányszerűen kerülnek bele, amikor a bizottság megvan a fordítással, vagy amikor egy új kódex, esetleg szabálykönyv megjelenik. A bevittet mindenképpen ellenőrzött formában kell végrehajtani a már most összetett szabályrendszer miatt. Valamint, a bizottságnak a feltöltést tesztelni is kell mielőtt használatra továbbítjuk.

Az eddigi tapasztalatok szerint, a seregek kialakítása mind lebontható kliccelős műveletekre. Ezt támasztja alá, hogy fizikálisan is reprezentálva vannak a seregek, és a kódexek is limitált opciókkal látják el a játékosokat. Kötött lehetőségek vannak kötött következményekkel, nincs semmi félmegoldás vagy skálázás.

A sereget érintő minden információnak rendelkezésre kell állnia, mivel a játékosok gyakran nem ismerik ellenfeleik sajátosságait, és a bizalom nem megoldás.

Jó, ha könnyedén megosztható a másik féllel minden, ami kérdéses lehet. A végterméket ki kell tudni nyomtatni papírformában, hogy ki lehessen osztani a játékosoknak. Mint ahogy szeretnénk megszüntetni a sokáig tartó szabálykönyv böngészést, úgy azt sem szeretnénk, hogy a szervező számítógépénél álljanak sorba az emberek vitás kérdések esetén.

Összegezve:

- Eseményeken szabad felhasználás
- Nem kell felhasználói feltöltési opció
- Gombos rendszer, nem kell gépelni
- Minden adat legyen elérhető
- Nyomtathatóság

#### 4. Tervezés

Annak ellenére, hogy kérésre fázisokban lesz a megvalósítás, a hátsó adatszerkezetet egészében kellett megtervezni, hogy mindennek meglegyen a helye és kapcsolatrendszere, amint eljön a fázisa.

Az adatszerkezetet, ami tartalmazza a szabályokat és az egyes kódexek adatait, úgy kellett kialakítani, hogy a jelenlegi összes eltérő forrás fogadására készen álljon. Magát a játék felépítését is elemeire kellett bontanom ahhoz, hogy kiderüljön, vajon hogyan építette fel a játékot a fejlesztő.

A tárolt adatok és a felhasználó által épített seregek adatai között van egy logikai gát. Míg a tárolt adatok egy központ nélküli egyenletes háló majdnem mindent mindennel összekötő kapcsolati rendszer, addig az épített seregek egy hierarchikusan felépített leképezése a korábbiak.

Játék technikailag az egyik egység a unit (osztag). Unitokat aktiválunk, annak tagjai egyszerre mozognak, lőnek vagy harcolnak. Egy unit lehet egy vagy több tagú, élőlény vagy gép. A

tagjai között lehetnek eltérő fegyverek vagy felszerelések. Vannak olyan egy fős unitok, amelyek csatlakozhatnak egy másik unitba, így unitban unit is lehet. Ezen megfontolásból az aktuálisan épített sereg alapegysége a unit.

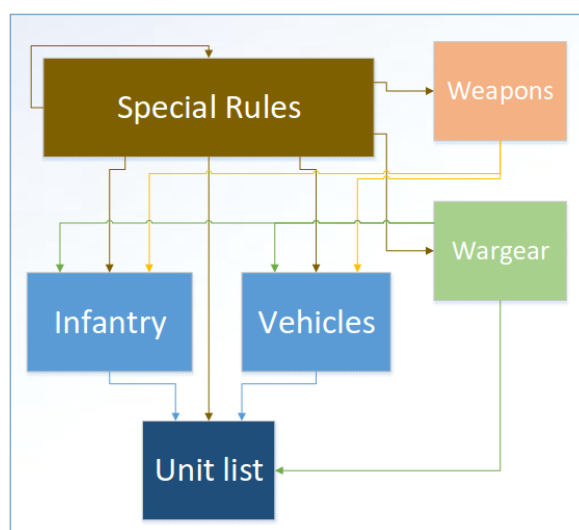
A unitok modellekből épülnek fel, és egy modell egy unitba tartozhat egyszerre, de ugyanaz a modell több unitban is lehet opció. A modelleknek az alaptulajdonságaik fixek, de unitonként kaphatnak eltérő felszereléseket, bónuszokat, amitől végeredményben már nem azonosak.

Vannak úgynevezett Special Rules (Egyéni Szabályok), amik bármihez rendelhetőek. Ezek a kiemelt szabályok adják a mozgatórugóját minden elemnek a játékban. Egyes felszerelések effektív csak egy extra képességet adnak egy osztagnak felszerelés formájában. Olyan is előfordul, hogy egy egyéni szabály, több egyéni szabály gyűjtő elnevezése.

Terjedelmi korlátok miatt csak néhány, az általam legfontosabbnak vélt terv kerül bemutatásra.

#### 4.1. Egyéni szabályok

Az egyéni szabályokat minden eddigi elemhez hozzá kell tudni rendelni, nem kötött számosságban. Vannak olyan egyéni szabályok, amik egyéb egyéni szabályokat tartalmaznak. Mint a felszereléseknel, itt is olyan az adat minősége, hogy gyakran a szöveges részt kell a játékosoknak értelmezni, és van egy kisebb porció, amelyet lehet számszerűsíteni módosítók formájában.

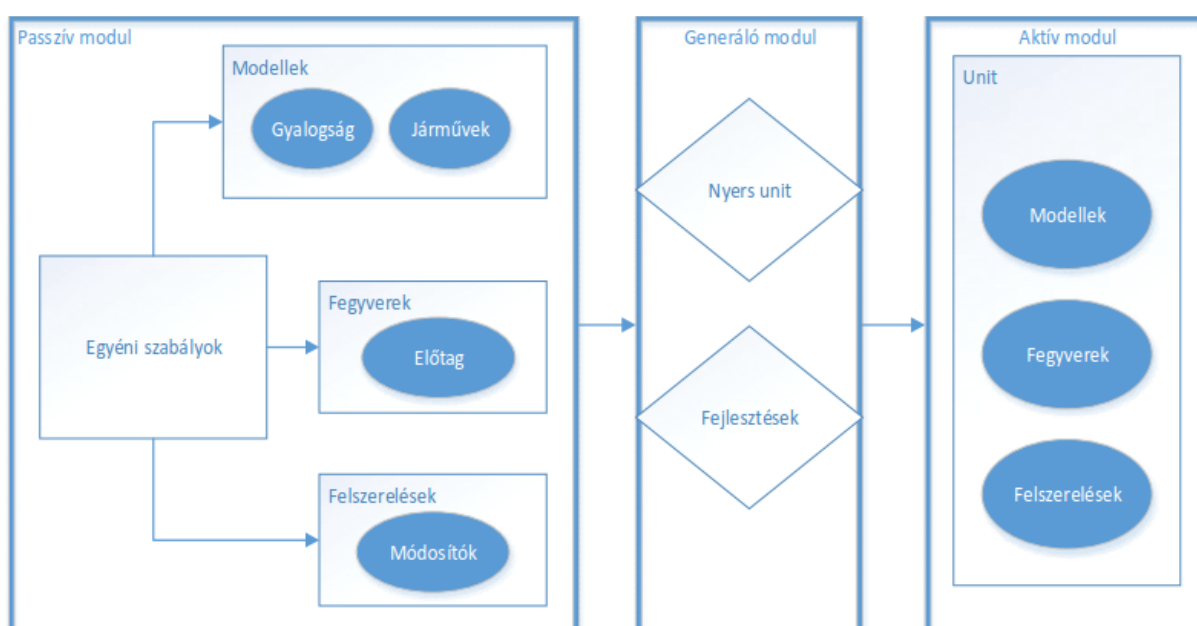


1. ábra: Egyéni szabályok kapcsolat rendszere

#### 4.2. Aktív és passzív adatok összekapcsolása

Miután megvan, hogy a felhasználó milyen formában végzi a seregépítést, és hogy a kódexek és szabálykönyvek adatait milyen formában is kell eltárolni, a következő logikai nézőpont a program végrehajtási státuszai, amelyek összekötik a kettőt.

A felhasználó egy hierarchikus rendszerben építi a seregét és a tárolt adatok egy hálórendszert alkotnak. A felhasználói vég az aktív rendszer, hiszen azt kell tudnia változtatni és építeni. A tárolt hálórendszer a passzív rendszer, mert abban semmilyen változás nem végezhető a folyamat alatt. A passzív rendszer elemeit felhasználva kell kialakítani az aktívát, és ennek a kialakításnak a módszerét, limitációját tartalmazzák a kódexek.



2. ábra: Adatmodell moduljai

## 5. Megvalósítás

A fentiekben megtervezett program, a következőképpen lett megvalósítva. Az adatszerkezetet egy MySQL adatbázis tárolja, a megjelenítését és felhasználói műveletek végrehajtását HTML+PHP végzi el, előforduló JavaScript felhasználásával.

## 5.1. *Adatbázis*

Az adatbázis elérése huzalozva van a php-be felhasználói adatokkal együtt. Az előkészített kapcsolat szerint. Mivel autentikációt nem vizsgálunk, a felhasználót teljes jogkörrel láttam el a felhasználás során.

### 5.1.1. Az adattáblák

Terjedelmi korlátok miatt csak néhány, véletlenszerűen kiválasztott adattábla kerül részletezésre.

- `conditions`: Az elvégzendő ellenőrzéseket tartalmazó relációs tábla. Ebben alálható, hogy milyen elemre, milyen ellenőrzést kell végezni a bemeneti paraméterekkel együtt. Nincsenek kapcsolatai.
- `infantry_type`: Link tábla, az `infatnry` és a `model_type` között. Az egyes gyalogsági modellekhez egy vagy több típust rendel a szabályok szerint.
- `issue`: Kiadványokat tartalmazó tábla. Itt találhatóak a hivatalos kötetek és a magyarra fordítottak is, hogy később lehessen választani változatot.
- `model_type`: A modellek típusai megkülönböztetik a modellt az alap típustól. Eltérő egyéni szabályokkal, tulajdonság módosítókkal ruházzák fel az érintett modellt.
- `vehicle_type`: Link tábla, a `vehicles` tábla és a `model_type` tábla között. Az egyes jármű modellekhez egy vagy több típust rendel a szabályok szerint.
- `vehicles`: A járművek alap tulajdonságait tartalmazó tábla. Az `inf_issue` mező idegen kulcsa, az `issue` tábla `i_id` egyedi kulcsához. Kódex verziószám váltáskor változhatnak a modell tulajdonságai és elérhetőnek kell lennie a régebbi változatoknak is.
- `wargear`: A létező felszereléseket tartalmazó adattábla. A `wg_issue` idegen kulcsa az `issue` tábla egyedi azonosítójának.
- `wargear_kit_raw`: Link tábla, a `unit_raw` és a `wargear` tábla között. A nyers unitok `wargearjeit` tartalmazza.

## 5.2. *A program*

Az adatbázis műveletek végrehajtását és a kezelőfelület megjelenítését PHP végzi HTML és JavaScript beágyazásával. Ez egy stabilan fejleszthető környezet, ami eléggé elterjedt ahhoz, hogy a jövőben más folytathassa a projektet a szabályok változása során. Továbbá eléggé



elterjedt ahhoz is, hogy minden böngésző probléma nélkül tudja kezelni és a felhasználóknak ne kelljen semmilyen új programot telepíteniük.

### 5.2.1. A kezelőfelület

Elsődleges célkitűzésem az egyszerűség volt. Ne kelljen senkit órákon át tanítani a program használatára. Amennyiben magát a játékot ismeri, a felület legyen magától értetődő. A második szempont, a felhasználói hibalehetőségek teljes kizárása. Maga a szabályrendszer is mederben tartja a lehetséges opciókat a játékosok számára, így megoldható volt, hogy lényegében csak irányított műveletekre korlátozzam a felhasználót. Egyetlen helyen kell saját adatot beadni, amikor a pontértéket határozza meg a sereg számára. Az összes többi művelet a program által felkínált választás, gombok formájában.

Az épített sereg hierarchikus felépítése, és közösségi vélemények alapján célszerűnek találtam, ha az egész seregépítés egyetlen oldalon látható folyamatosan.

Ilyen módon a felhasználók könnyebben átlátják, hogy mijük van és mijük lehet. A frakciót és pontszámot is hasznos folyamatosan megjeleníteni informatív jelleggel.

Amikor a felhasználó végzett, ki kell tudja nyomtatni a sereg listáját a hasznos információkkal, de szerkesztő opciók nélkül összegezve.

Ezek alapján három felülettel fog találkozni a felhasználó: Frakció választás, sereg építés és nyomtatás.

### 5.2.2. A felhasználó folyamatai a programban

A felhasználó elsőként az index.php-vel találkozik, ahol a sereg frakcióját és pontértékét tudja megadni. A frakció egy legördülő menü, amiből választhat, a pontértéket szabadon tudja megadni, numerikus formában. A "Rendben" gombbal tovább léphet.

Következő képernyője egy frameset, a main.php. Itt láthatja a fenti részben, a korábban megadott adatokat, a számára létrehozott adattábla azonosítóval együtt. Első megjelenítésre, csak az előkészített adatok jelennek meg számára. Ezek a frakcióban létező unitok, beosztás szerint. A beosztásokat gombok segítségével változtathatja és az alattuk található lista frissül.

Az "Felvétel" gombbal a kiválasztott unitot a program felveszi a seregbe és jobbra megjeleníti a sereg tartalmában.

A "Sereg felépítése" ablakban találhatóak a seregbe felvett unitok és az őket alkotó modellek. A "Törlés" gombbal eltávolítja az egész unitot, minden tartalmával együtt. A "Részletek" gomb megnyitja a unitot alkotó modelleket a "Unit összetétel" ablakban, és azok fejezetét a "Unit fegyverei" ablakban.

A "Unit összetétel" ablakban a "+" és "-" gombokkal módosíthatja a felhasználó a modellek létszámát. Itt láthatja az összes lehetséges modellt, ami előfordulhat a unitban.

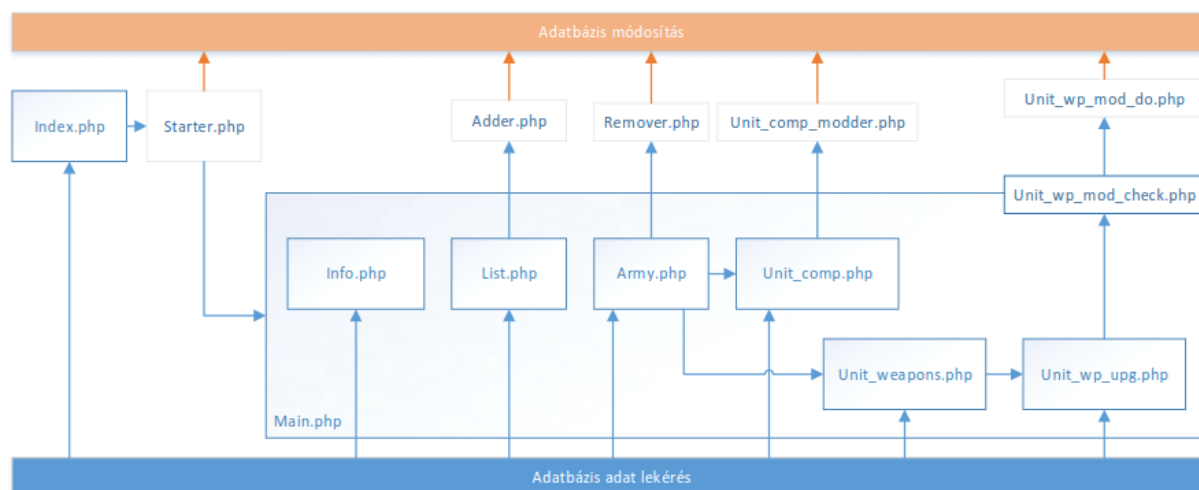
A "Unit fegyverei" ablakban láthatja, hogy az eltérő modell fajtáknál mennyi és milyen fegyverek vannak. A "Fegyverek" gombra kattintva a lehetséges fegyver fejlesztések megjelennek a "Választható fegyverek" ablakban. Ott a "+" és "-" gombokkal vehetik fel vagy törölhetik a kiválasztott fegyvereket. Olyan esetben, ha valamilyen már meglévő fegyver helyére lehet csak felvenni a kívánt fegyvert a felhasználó kiválaszthatja a felkínált lehetőségek közül, hogy melyiket kívánja cserélni.

Miután a szerkesztéssel végzett a felhasználó, a jobb fenti sarokban található "Nyomtatási kép" gomb segítségével, egy popup ablakban nyomtatható formában tálalja az egész sereget, minden elérhető szabállyal és tulajdonsággal együtt.

### 5.2.3. A program folyamatai

A felhasználói folyamatot a háttérben a program hajtja végre és ellenőrzi, hogy minden a játék szabályai szerint történjen. A felhasználó számára nem látható műveleteket a felület és az adatbázis között gyakran több lépcsőben, több file-on keresztül valósítja meg a program.

Általánosságban elmondható a program kódról, hogy van egy megjelenítő és egy módosító réteg. A megjelenítő réteget látja és használja a felhasználó, míg a módosító réteg változtatja az adatbázist. A megjelenítő réteg is tart kapcsolatot az adatbázissal, de csak és kizárólag select-eken keresztül.



3. ábra: A program szerkezet modularitása

A fenti ábrán nincs jelölve, de a módosító file-ok is kérnek be adatot az adatbázistól, és sem az, hogy milyen változtatás után, melyik file-nak kell frissítenie.

#### 5.2.4. A folyamat file-onkénti részfeladatai

Minden file-nak megvan a szerepe a folyamatban, amit ellátnak saját kereteiken belül. Vannak olyanok, amelyek nem szerepelnek közvetlen a felhasználói folyamatban. Ezek a file-ok függvényeket tartalmaznak, amiket a többi file hívhat meg. Céljuk a rendezettség és átláthatóság.

#### 5.2.5. A file-ok közötti adatáramlás

A file-ok a fent bemutatott feladataikat különböző féle adatok ismeretében kell, hogy elvégezzék. A file-ok, ahogy elváltak a megjelenítő és a módosító modul, úgy kell egymás között is adatot cserélniük. Általánosságban a megjelenítő modul adatbázis adatokat kér be magának és felhasználói adatokat ad át a módosító modulnak. A módosító modul pedig a kapott adatok feldolgozása után frissítésre kényszeríti a megjelenítő modul érintett elemeit.

A program adatait, amivel dolgozik 7 kategóriába soroltam:

- Bekért: A felhasználotól származó információ
- Továbbított: A file-ok között továbbadott adat a feladó oldaláról.
- Fogadott: A file-ok között továbbadott adat a fogadó oldaláról
- Session: Olyan adatok, amelyeket a file-ok nem közvetlen egymásnak adnak át, hanem session-be teszik ki. A fogadó oldalról ez Fogadott adat.

- Teremtett: A program által előállított adat.
- Módosító: Az adatbázisnak tárolásra küldött adat.
- Lekérdezett: Az adatbázisból származó adat

## 6. Konklúzió

A célkitűzésben megfogalmazott fejlesztés önálló tevékenység formájában valósult meg, amely során felismerhetővé vált a szakmai fejlődés lehetősége is, amely még inkább megjelenik a programozás tárgyak körében. Az ilyen jellegű tevékenységek – különösen az egyetemi oktatásban – egyenrangú jelenhetne meg az iskolai tanulásban mivel így a tanulást elősegítő motivációk felerősödhetnek és a differenciálás is széleskörűen megvalósulhatna.

Az elvégzett fejlesztés tesztelése során vett eredmények alapján megállapítható, hogy az adatmodell megfelelőnek bizonyult, minden fontos információt tartalmazott és minden fontos információ visszavezethető. A klubok folyamatosan tesztelik a rendszert és küldik a visszajelzéseket. Az élő rendszerben már 3 faj és 14 kódex van a jelenleginek megfelelő szinten feltöltve. A jövőben be kell építeni majd a hiányzó részeket is, mint a felszerelések és a modell típusok. A logikai előkészítésük már megvan, és a tényleges adatbevitel után megkezdődhet a program továbbfejlesztése is. Az első magyarra fordított részek még váratnak magukra, de hiszem, hogy integrációjuk minimális fejlesztéssel megoldható. A tervek szerint, egy szűrő réteg kerül majd a mostani felé, ami csak bizonyos issue készletekből engedni válogatni a programot, így szabálycsaládokat létrehozva a klubok számára. Szélsőséges tesztek le tudták lassítani a működését az online változatnak, de összeomlást nem okoztak. A fejlesztői környezet tökéletes volt a projekthez. A további fejlesztéseket is lehetővé teszi akár más fejlesztő számára is. Az adatbázist igény szerint könnyen át telepíthető.

## Irodalomjegyzék

Buckley P., Lee P. (2018). The impact of extra-curricular activity on the student experience. *Active Learning in Higher Education*.

Carter, M., Harrop, M., & Gibbs, M. (2014). The roll of the dice in Warhammer 40,000. *Transactions of the Digital Games Research Association*, 1(3).

Games Workshop Ltd (2015). Codex Adeptus Astartes – Dark Angels. *Games Workshop*, Nottingham.

Meloni J. C., Bálint G., László T. (2003). *Tanuljuk meg a MySQL használatát 24 óra alatt: 24 egyszerű, egyórás lecke*. Kiskapu.

Judit, P. K. (2014). Tanórán kívüli iskolai és iskolán kívüli programok a tehetséggondozásban, Magyar Tehetségsegítő Szervezetek Szövetsége.

Katona, J. et al (2016). Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based brain-computer interface. In *2016 7th IEEE international conference on cognitive infocommunications (CogInfoCom)*, 251-256

K. J., K. A. (2015). EEG-based Computer Control Interface for Brain-Machine Interaction. *International Journal of Online Engineering*, 11(6).

Ullman, J. D., Widom, J., Cserges, E., Csizmazia, B., & Gyenizse, P. (1998). *Adatbázisrendszerek: alapvetés*. Panem.

Sziladi et al (2017). The analysis of hand gesture based cursor position control during solve an IT related task. In *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 413-418.

Zandstra, M., István, O., László, T., & László, R. (2005). *Tanuljuk meg a PHP 5 használatát 24 óra alatt*. Kiskapu.