

**Szegedi Tudományegyetem**

**Informatikai Tanszékcsoport**

# **SZAKDOLGOZAT**

**Hubert Viktor**

2013

**Szegedi Tudományegyetem**

**Informatikai Tanszékcsoporth**

**Webes csoportos szerepjáték segítő rendszer  
(RPGCraft)**

**Készítette**

**Hubert Viktor**

Programtervező Informatikus hallgató

**Témavezető**

**Havasi Ferenc**

Szeged

2013

## **Feladatkiírás**

A rendszer feladata annak elősegítése, hogy szerepjátékot tudjanak játszani egymástól távol lévő emberek is fájl/térkép megosztással, kör és privát üzenetekkel, karakterlapok kezelésével - amelyet a játékmester és az illető játékos módosíthat.

## Tartalmi összefoglaló

Szakedolgozatom témája egy webes alkalmazás fejlesztése amely szerepjátékok kezelését könnyíti meg egymástól távol lévő emberek számára chat szolgáltatással, fájlok megosztásával, karakterlapok kezelésével, egyéb kényelmi funkciókkal.

A rendszer használatához elegendő egy böngésző. A rendszer leíró nyelve a HTML.

A backend megírásához PHP-t, a frontenden pedig JavaScriptet – azon belül is a jQuery keretrendszert – használtam. Emellett a stílusozást CSS-el és LESS-el valósítottam meg.

A rendszer saját template-et – sémát – használ, ami a felhasználóbarát URL-ért, többnyelvűségért és egyéb optimalizálásokért felel.

A rendszert egy weboldalon keresztül lehet elérni. A regisztráció ezen keresztül lehetséges (Csak regisztrált felhasználók játszhatnak.), SEO szempontokat is figyelembe vesz és megkülönbözteti a látogató, regisztrált és admin felhasználókat.

Elért eredmények:

- Szerepjátékok játszhatóak egy interfészen keresztül,
- A karakterlapok kezelhetők a kalandmester vagy a tulajdonos játékos által,
- Karakterlap Fejlesztő Eszköz (Ehhez dokumentáció, online validáló eszköz elérhető).
- Chat lehetősége a játékon belül,
- Kockadobás lehetősége,
- Fájl- és térképmegosztások,
- Elkülönített jogosultságok: látogató/felhasználó/admin/kalandmester,
- Regisztráció és belépés (aktivációs e-mail, elfelejtett jelszó),
- Barátok bejelölése, felvétele,
- Játékok indítása kalandmesterként, játékfelkérések küldése barátoknak,
- Többnyelvűség.

Kulcsszavak: HTML, XML, JSON, CSS/LESS, PHP, MySQL, JavaScript, jQuery, AJAX

## Tartalomjegyzék

Feladatkiírás.....	3
Tartalmi összefoglaló.....	4
Tartalomjegyzék.....	5
<b>1. Bevezetés.....</b>	<b>7</b>
<b>2. Szükséges háttérismeretek.....</b>	<b>9</b>
2.1. SGML.....	9
2.2. HTML.....	9
2.3. AJAX.....	10
2.4. XML.....	11
2.5. CSS/LESS.....	11
2.6. PHP.....	12
2.7. MySQL.....	13
2.8. JavaScript, jQuery.....	13
<b>3. Az adatbázis.....</b>	<b>15</b>
<b>4. A rendszer.....</b>	<b>18</b>
4.1. Weboldali keret.....	18
4.1.1. Figyelmeztetések.....	19
4.1.2. Felkérések.....	20
4.1.3. Ismerősök.....	20
4.1.4. Profil módosítása.....	20
4.1.5. Játékok kezelése.....	21
4.1.6. Fejlesztők (Karakterlap Fejlesztő Eszköz).....	22
4.2. Játék színtér.....	26
4.2.1. AJAX lekérések.....	27
4.2.2. Szinkronizálás.....	28
4.2.3. Chat.....	29
4.2.4. Kockadobás.....	29
4.2.5. Csatlakozott játékosok.....	29
4.2.6. Térképek.....	30
4.2.7. Bejegyzések.....	31
4.2.8. Karakterlap.....	33

<b>5. Karakterlap dokumentáció.....</b>	<b>35</b>
5.1. Magyarázat a kódokhoz.....	36
5.2. XML dokumentáció.....	38
Összefoglaló.....	41
Irodalomjegyzék.....	42
Nyilatkozat.....	43
Melléklet.....	44
Köszönetnyilvánítás.....	45

## 1. Bevezetés

Szakdolgozatom célja a szerepjátékok virtualizálása, és kiegészítése hasznos funkciókkal. Ezt továbbgondolva egy olyan eszköz kifejlesztése volt a célom, ami egy online közösség segítségével is bővíthető lehetne.

### **Melyek a rendszer által feldolgozni kívánt szerepjátékok?**

Ezek a játékok alapvetően verbális jellegűek, jellemzően 4-5 ember alkotja, akik közül egy a mesélő – Vagy más néven Kalandmester. (KM a hátralévő fejezetekben.) –, míg a többiek a játékosok. Közösén, szóban elbeszélve egy fantáziavilágban történt eseményeket élnek át, mindenki az elképzelt karakterével. A KM szerepe a világban történt dolgokat elmesélni, formálni az eseményeket, irányítani a nem játékos karaktereket. Ő helyezi el a játékosokat a világ egy pontján, ahonnan a történet kezdődik, elmeséli a látottakat, válaszol a játékosok kérdéseire és leírja, hogy egy-egy cselekedetre mik az őket körülvevők reakciói.

A mesélt játéktól függően mindebben legtöbbször a KM rendelkezésre áll egy előre megírt világ; annak történelme, kultúrája, a benne élő lények.

Az egyik legfontosabb eleme ezeknek a játékoknak a kockadobás, amit jellemzően 4, 6, 8, 20 oldalú kockákkal – Melyeket d4, d6, d8, d20-ként szokás említeni ezekben a játékokban. – szoktak játszani. Különbő eseményekhez, harcokhoz használják a játékosok, vagy a KM, így az egyik legnagyobb befolyással rendelkező elem. Mivel a dobás eredménye véletlenszerű, a játékok furcsa fordulatokat vehetnek, így színesítve azt.

Például kockát kell dobni valahányszor egy játékos megsebzí ellenfelét és ki kell számolni mi a sérülés mértéke, vagy ugyan így kockadobással eldönthető, hogy egy blöff sikeres-e.

A szerepjátékok típusai, rendszerei széles skálán mozognak, változatos harcrendszerek, világok közül válogathatunk.

### **Pár példa a teljesség igénye nélkül**

- *MAGUS.*
- *DND.*
- *Shadowrun.*
- *Vampire,*
- *stb.*

Ugyan akkor nem muszáj ragaszkodni az előre megírt játékokhoz. Ha valaki elég találékony, alkothat saját játékot a kedvenc történelmi eseményével kapcsolatban, egy kedvelt regényből, vagy akár sorozatból.

Alapvető probléma, ha egy adott játékos a többiek számára ismeretlen, vagy nem kedvelt játékot szeretne játszani. Egy új, ismeretlen játék rendszerének elsajátítása sok időt emészt fel, másrészt ha olyasvalakivel szeretnénk játszani, akivel nem tudunk személyesen találkozni, a lehetőségeink korlátozottak.

### **Megoldások**

- Fizetős rendszerek használata,
- Valamilyen hangátviteli program használata.

Ezeknek a megoldásoknak azonban komoly hátrányaik vannak. A fizetős rendszerek azon túl, hogy fizetősek, sokszor nehezen konfigurálhatóak, és csak korlátozott mennyiségű játékot támogatnak.

Megoldást nyújthatnak konferenciahívások, de nagyban csökkenti a játékból fakadó varázst. A játékmenet ilyenkor akadozó, általában véve nem népszerű módszer.

### **Szakedolgozatom által kínált megoldás**

Online rendszer írása, melyet bárki szabadon használhat.

Nem kell hozzá más, csak egy böngésző és egy gyors regisztrációt követően már válogathatunk a játszani kívánt játékok között.

### **A rendszer nagy erősségei**

- Ingyenes,
- Új játék típusok készíthetők regisztrált játékosok által (A felvitel ellenőrzött.),
- Bárki indíthat kalandot, behívhatja az ismerőseit,
- Játékon belül kényelmi funkciók, amik a játék további megkönnyítésére valók. (Például: Chat, fájl- és térképmegosztás.)

Ily módon a felhasználók is részét képezik a bővítéseknek, rajtuk keresztül egyre több játék típus kerülhet fel.

A rendszernek az **RPGCraft** nevet adtam. Ez a név az egyéni karakterlapok készítésének fontosságát emeli ki, amely a legnagyobb erőssége.



## 2. Szükséges Háttérismeretek

### *Leíró nyelvek*

A különböző leíró nyelvek segítségével tudjuk az információt a számítógépeknek is kezelhető formában tárolni. Hogy a különböző leíró nyelvek szintaxisát hogy kell értelmezni, DTD-ben (Document Type Definition, Dokumentum Típusdefiníció) határozzák meg.

Itt az SGML és HTML leíró nyelvekre térek ki.

### 2.1. SGML

Az SGML (Standard Generalized Markup Language, szabványos általános jelölőnyelv) egy ISO szabványos jelölőnyelv dokumentumformátumok leírására.

Az SGML egy absztrakt szintaxist biztosít, amit sokféle alkalmazásban használhatunk. A szabványos szintaktika lehetővé teszi, hogy az ilyen formátumú dokumentumokat egy általános célú értelmezővel (parser) könnyen beolvashassuk, írassuk vagy formailag ellenőrizhessük. SGML-ben a jelölések (tag) jelentése nincs meghatározva, ez mindig az SGML-t használó alkalmazás feladata (például a HTML-ben, ami az egyik legismertebb SGML alkalmazás, a jelöléseknek már konkrét jelentésük van, és a jelölések értékkészlete véges).

### 2.2. HTML<sup>[1]</sup>

#### HyperText Markup Language

Az általam is használt leíró nyelv. A HTML úgynevezett HTML elemekből áll, amelyek nyitó és záró „tag”-ekből állnak. (pl.: <html></html>, bár nem minden esetben szerepel különálló záró tag, például: <img>, <br>)

Ezek a HTML elemek a weboldalak építőkövei. A böngésző feladata, hogy ezeket értelmezze majd a tartalmat a megfelelő formában jelenítse meg.

A HTML elemek az adatok struktúrált megjelenítésére használatos elsősorban. (Bár vannak stílusformázó elemek is, mint például a „<font>”, ezek használata nem javasolt.) A tartalom így elkülöníthető fejlécekkel, paragrafusokkal, sortötéssel, listákkal, linkekkel, idézetekkel és még sok egyéb mással.

Az alapvető HTML felépítés:

```
<html>

  <head>

    <title> „Hello World!” In HTML. </title>

  </head>

  <body> Hello World! </body>

</html>
```

(Ez egyúttal tekinthető egy nagyon egyszerű weboldallaknak.)

A fejrészbe (<head>) kerülnek a információk az oldalról, scriptek (<script> elem), stílusleíró kódok (CSS, <link> elem), metainformációk (<meta> elem) a böngésző számára a megjelenítésről.

A törzsbe (<body>) kerül a megjelenítendő tartalom.

### 2.3. *AJAX*<sup>[2]</sup>

Aszinkron JavaScript és XML. (Asynchronous JavaScript and XML.)

Az AJAX lényegében összefüggő webtechnológiák összessége, amit aszinkron webalkalmazások létrehozására használnak.

Segítségével ezek az alkalmazások képesek adatok küldésére és fogadására egy szervertől aszinkron módon (azaz a háttérben) anélkül, hogy ezzel a megjelenített oldalon változtatnánk.

Az adatokat az úgynevezett XMLHttpRequest (XHR) objektummal érhetjük el. Az elnevezéssel ellentétben nem kötelező XML-t használni (Mint ahogy erre itt is lesz bőven példa: az itt használt AJAX lekérések JSON-ban kezelik az adatokat.), és nem kell az aszinkron lekéréshez sem ragaszkodni. (Ekkor az oldal letöltése is megáll addig, míg az AJAX lekérés nem végez, így ilyen lekérésre ritkán van szükség, itt minden lekérés aszinkron lesz.)

Összességében az AJAX nem egy különálló technológia, hanem több együttes alkalmazása. Segítségével elkerülhető az oldal újratöltése.

## 2.4. XML<sup>[3]</sup>

Míg a HTML adatok megjelenítésére szolgál, úgy az XML az adat továbbítására, tárolására úgy, hogy az mind ember, mind gép számára érthető formátumú legyen.

XML (EXtensible Markup Language) egy olyan nyelv, amelynek elemei nem előre definiáltak, így saját struktúrát lehet alkotni benne, amely a leginkább megfelel céljainknak.

Esetemben az XML karakterlap struktúrák tárolására szolgál, amelyet aztán több algoritmus dolgoz fel.

A rendszer a felhasználó által megírt XML-t értelmezi egy PHP-fájllal (SimpleXML-t használva, a PHP 5-ös verziójától elérhető kiegészítést.), ahol az adott tagek jelentik a táblák elnevezéseit, mezőit, bizonyos attribútumok megadásával pedig a típus is meghatározható.

(Az ezzel kapcsolatos konkrét dokumentáció a szakdolgozat végén található.)

## 2.5. CSS/LESS

Mint ahogy azt említettem, a HTML az adatok struktúrázására szolgál, nem pedig a megjelenítés megszabására. Azonban a technológia megjelenést követően hamarosan igény jelentkezett arra, hogy a megjelenítés stílusa is megszabható legyen.

Erre külön elemek jelentek meg (Mint például: <i>, <b>, <font> stb.), de ezek helyett is szükség volt valamire, ami dinamikusabbá teheti a megjelenítés megszabását.

Erre ad megoldást az úgynevezett CSS<sup>[4]</sup> (Cascading Style Sheet), azaz a stíluslap. Segítségével a HTML elemekhez úgynevezett class-okkal megadhatjuk azok megjelenítését.

A „Cascading” szó a technológia azon tulajdonságából ered, hogy egy elemhez több stílus is megadható, amelyek tulajdonságai össze is adódhatnak. Ha egy adott tulajdonságot több class is megad, akkor a leg szorosabban kötő class tulajdonsága lesz alkalmazva.

Példa

### HTML

```
<div class="highlight"></div>
```

### CSS

```
.highlight {  
    font-weight: bold;  
}
```

A stíluslapok megadásának több módja is van.

1. Alapvetően jó fejlesztői hozzáállás, ha a stíluslapokat külön '.css' kiterjesztésű fájlokban helyezjük el, majd ezeket linkeljük a megfelelő oldalakon.
2. Megadható a HTML fejrészben is, `<style></style>` tagek között.
3. Az adott elemhez is megadható, `style=""` módon. Bár ez utóbbi nem javasolt, mégis előfordulnak helyzetek amikor erre van szükség.

A **LESS**<sup>[5]</sup> a CSS nyelv kiegészítése. Egy dinamikus stíluslap generátor, ami kiegészíti az alapvető CSS tulajdonságokat úgy, hogy az jobban hasonlítson egy programozási nyelvre.

Megadhatók benne változók, matematikai műveletek, függvények, így még könnyebbé téve a fejlesztői munkát. A böngésző szempontjából nincs változás, hiszen az eredmény ugyan úgy CSS lesz, amit aztán a HTML kódhoz illeszt.

A LESS használatának több módja is létezik: kliens- , szerver oldali értelmezés, vagy elő fordítás a fejlesztés során.

Kliens oldali értelmezéskor a .less fájl nincs előre css-re lefordítva, helyette mellé a less értelmezője van linkelve. (Tisztán JavaScript.) Ez a script az oldal megjelenítésekor lefordítja a less fájlt css-re, így a böngésző csak ez utóbbival kell foglalkozzon. Óvatosan kell ugyanakkor bánni ezzel a megoldással, mert nagy fájlok esetén a fordítás ideje észrevehető lehet, főleg lassabb internetkapcsolattal rendelkezők számára.

Szerver oldali fordítás esetében a fordítást a szerver végzi el.

Végül, a módszer amit a itt leírt rendszer is használ: a fejlesztés alatt előre fordítás. Ekkor a fejlesztés LESS fájlokkal történik, de az oldal beüzemeléskor azok előre lefordításra kerülnek CSS formátumba.

## **2.6. PHP<sup>[6]</sup>**

PHP: Hyper-text Preprocessor.

A PHP egy szerver-oldali nyelv, amit elsősorban webfejlesztéshez terveztek.

A PHP kódot a webszerver értelmezi, majd hozza létre a megjelenítendő oldalt. Használhatóak benne a programozásban megszokott függvények, így: változók, tömbök, ciklusok, függvények, objektumok stb.

Gyengén típusos nyelv, nem követeli meg, hogy típusokat deklaráljunk. Így nem kell foglalkozni pl.: egy szám méretével, csak megadunk egy változónevet és eltároljuk. A többit elvégzi a PHP.

Lehetőség van a HTML-be ágyazni: „<?php ?>”, vagy „<? ?>” elválasztó jelek közé, pl:

```
<?php echo „Hello World!”; ?>
```

Ez lehet több soros, vagy csak soron belül egy szó kiíratása, ezek a jelek mondják meg, hogy php kód következik, amit a szervernek értelmeznie kell.

A rendszer nagy része PHP kiterjesztéssel íródott többek között az adatbázis kezelése is PHP-n keresztül valósult meg.

## 2.7. *MySQL*<sup>[7]</sup>

Nyílt forráskódú RDBMS, azaz relációs adatbázis. A standard SQL-nyelvet használja.

Az adatbázisban tárolódnak az adatok, és az első szerver-kliens kapcsolat is részben az adatbázison keresztül valósult meg.

## 2.8. *JavaScript, jQuery*

A JavaScript<sup>[8]</sup> kliens-oldali programozáshoz, DOM-manipulációhoz (Document Object Model) használatos. Segítségével a weboldalakat valós időben, újratöltés nélkül változtathatjuk. A HTML résznél említett tag-eket befolyásolhatjuk, azok tartalmát, attribútumait.

Ugyan úgy szerepelnek benne funkciók, változók, tömbök, objektumok, ezek segítségével doglozhatjuk fel a kívánt információt. Egy népszerű használata az úgynevezett AJAX. (Asynchronous JavaScript) Célja hogy a háttérben, újratöltés nélkül kérjen le új információkat, majd azokat a kliens-oldalon kezelje. Sok helyen használt ez a technika a weboldalon, legfontosabb rész a szerver-kliens kapcsolatnál. (Részletesebben kifejtem az elért eredményeknél.)

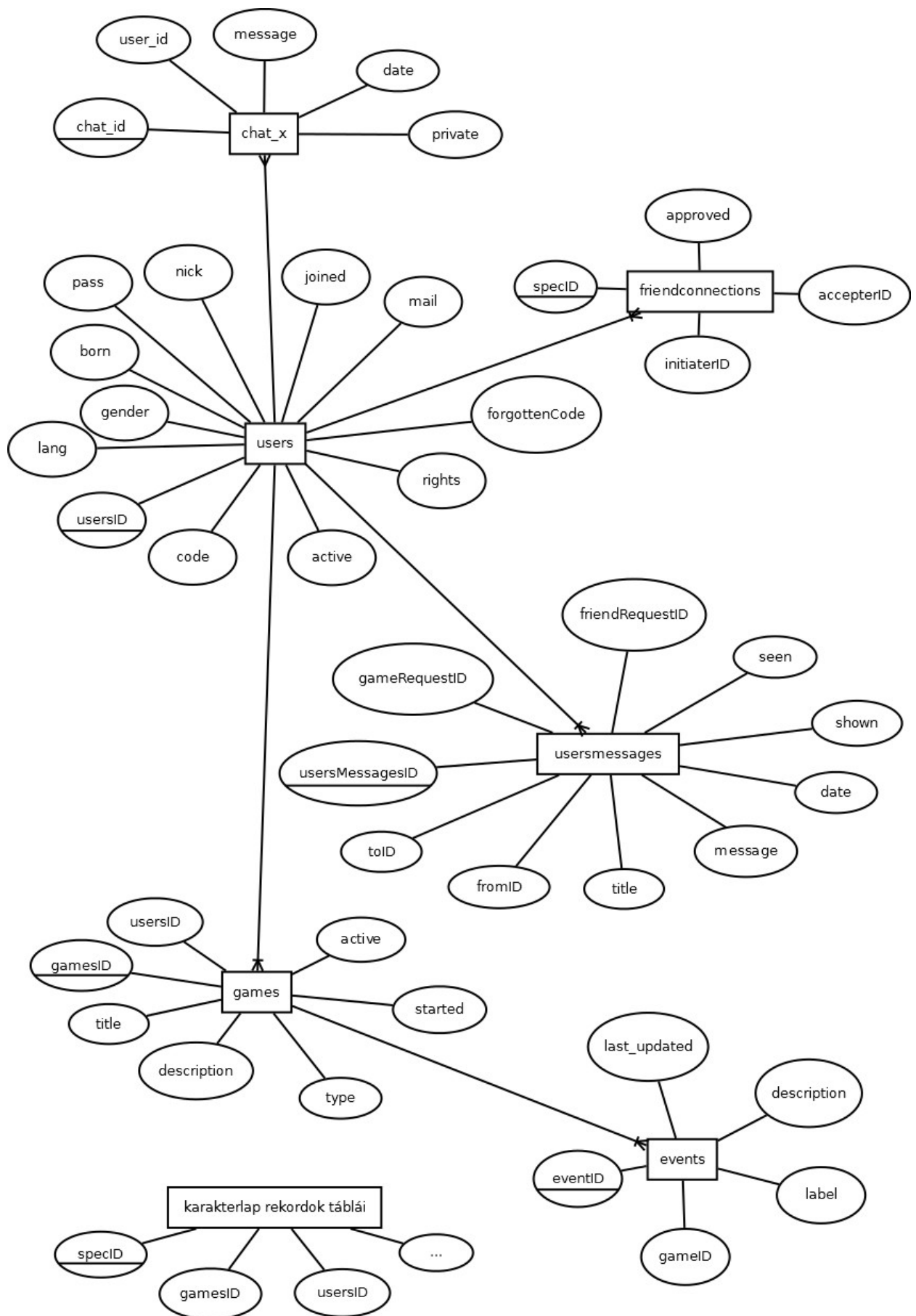
A böngészők közötti eltérések miatt viszont nehéz vele független, mindenhol helyesen megjelenő és működő kódot írni, illetve a programozás maga nehézkes.

A jQuery<sup>[9]</sup> könyvtár többek között erre ad megoldást. Segítségével a javascript programozás egyszerűbb és hatékonyabb. Sok beépített függvényt tartalmaz, ami javascript parancsok sorozatát hajtja végre úgy, hogy az böngészőfüggetlen legyen, és a scriptek folyamatos

kapcsolatot tartsanak a programozóval úgynevezett 'handlers' segítségével. Sok kényelmi, animációs szolgáltatása is van, mind a fejlesztés megkönnyítésére.

Beépített AJAX funkcióval és a handlersszel rendelkezik, aminek segítségével lekezelhetőek a hibaforrások, pl.: időtúllépések, megszakadt internetkapcsolat.

### 3. Az adatbázis



**users:** Itt tárolódnak a felhasználók adatai.

'*code*': az első belépéskor kapott aktivációs kód, amit regisztráció után e-mailben kapunk meg. Ha a megadott linken keresztül visszalátogatunk az oldalra, az 'active' flag igaz lesz, így a felhasználó beléphet.

'*forgottenCode*': új jelszó generálásához kapott kód. Ugyan úgy működik mint az előző, csak ezt az új jelszó generálásakor kapjuk. Ha a kapott linken keresztül visszalátogatunk az oldalra, egy új jelszót generál a rendszer.

A többi mező értelemszerű.

**usersmessages:** A felhasználókhoz érkezett üzenetek. A rendszer jelenlegi verziója üzenetküldésre nem, helyette barát felkérések kezelésére és játékfelkérésekhez használja a táblát.

'*seen*', '*shown*': Ezek a mezők, flag-ek attól függően állítódnak be, hogy az adott üzenetet a rendszer megmutatta-e a felhasználónak, illetve az ténylegesen bele is nézett-e az üzenetbe. (Utóbbi egy későbbi verzióban lehet használatos, ha tényleges, megnyitható üzenetekről is szó lehet.)

A többi mező értelemszerű.

**friendconnections:** A baráti kapcsolatokat tároló tábla. A benne lévő két ID alapján egyértelműen meghatározható. Ez határozza meg azt is, hogy egy játék kalandmestere kiket hívhat be a játékba.

'*initiatorID*': A felkérést küldő felhasználó ID-je.

'*accepterID*': A felkérést elfogadó felhasználó ID-je.

'*approved*': Ha a felkérést kapó felhasználó elfogadja azt, akkor válik ez a flag igazzá.

**games:** Egy játék elkészítésekor ide kerül új rekord rögzítésre.

'*type*': A játék típusa. Egy egész szám, amihez tartozó adatok egy ini fájlban kerülnek rögzítésre, mikor egy új játék a rendszerbe kerül.

'*active*': Ez a jelenlegi verzióban nem használt. Ez a flag árulta el, hogy egy játékot éppen játszanak-e, azaz aktív-e éppen. Biztonsági okból megmaradt.

A többi mező értelemszerű.

**events:** A játékon belüli bejegyzések. (Az *events* elnevezés a fejlesztés korai szakaszából adódik, amikor ténylegesen csak események rögzítésére volt fenntartva. Azóta a játékon belül nyelvtől függően az *entries* vagy *bejegyzések* néven található meg.)



Az itt szereplő mezők egyszerően adódnak. A bejegyzésekhez tartozhat cím és leírás, illetve hozzá csatolt képek.

**chat\_x**: Egy játékhoz tartozó chat-tábla, ahol x a játék ID-jét jelenti.

Minden játékhoz készül egy ilyen tábla a létrehozásakor.

**Karakterlap rekordok táblái**: Ezek a táblák változatos elnevezésűek és struktúrájúak, mivel az majdnem teljes egészében a karakterlapot leíró XML-től függ.

Három mező viszont minden táblához létrehozódik.

'*specID*': A táblán belüli rekord egyedi ID-je, amivel később hivatkozni lehet rá törléskor/módosításkor.

'*usersID*': A rekordot feltöltő játékos ID-je.

'*gamesID*': Annak a játéknak az ID-je, amelyhez a rekord tartozik.

'...': A többi mező a karakterlap XML-je alapján generálódik úgy, hogy az abból kinyert mezők mellé egy '*vis\_*' prefixxel ellátott pár is kerül, ami megszabja, hogy az adott mezőt más játékosok láthatják-e.

Így például, ha van egy '*gold*' mezőnk, ahhoz generál egy '*vis\_gold*' mezőt is. A játékos ezzel határozhatja meg, hogy megtekinthető legyen-e az aranytartaléka, vagy sem.

## 4. A rendszer

A rendszer két fő részből áll:

- Weboldali keretrendszer: Ebbe ágyazódik a játék rendszere. Itt lehet regisztrálni, játékhoz csatlakozni, vagy indítani egyet, kezelni az ismerősöket.
- A játék színtér: Itt zajlik a tényleges történetmesélés, karakterlapok kezelése.

### 4.1. Weboldali keret

A weboldal megkülönböztet látogatót, (regisztrált) felhasználót és admin, ezek mellett több nyelvet is támogat. (A fejlesztés alatt az angolt és magyart tartottam szem előtt. )

Az oldal első indításakor a látogatói felület fogad. Itt három fontosabb menüpont vár: Hírek, Regisztráció, Belépés, illetve letölthető a karakterlap régi és új verziója.

Minden oldalon nyelvet lehet váltani. (Magyar az alapértelmezett.) A nyelvváltás látható az URL-ben: angol nyelv esetén megjelenik a '/en/', magyar esetén a '/hu/'. Az alapértelmezett nyelvhez ez nem kötelező, így például a: '/hu/belepes/' alak ugyan azt az oldalt hozza be, mint a '/belepes/'. A rendszer tartja is magát ehhez, ha lehetőség van rá, egyszerűsít.

A használt URL-ek beszédeseek, a módszert 'url routing'-nak is hívják. Így könnyebb megjegyezni a címet és egyben keresőbarát is.

Példák: '/belepes/', '/regisztracio', '/game/' stb.

(Ahol a domainnév és gyökérkönyvtár mindezeket megelőzi.)

A regisztrációt egy e-mailben megkapott aktivációs kóddal tudjuk megerősíteni, majd ez után belépve elérhetjük a felhasználói felületet.

A felhasználói felületen belül újabb menüpontok jelennek meg.

- Figyelmeztetések,
- Felkérések,
- Ismerősök,
- Profil módosítása,
- Játékok kezelése,
- Fejlesztők (Karakterlap Fejlesztő Eszköz).

#### 4.1.1. Figyelmeztetések

A JavaScript alapértelmezett alert() és confirm() funkciói helyett egy saját jQuery plugint írtam, amit HValert-nek kereszteltem el. Így

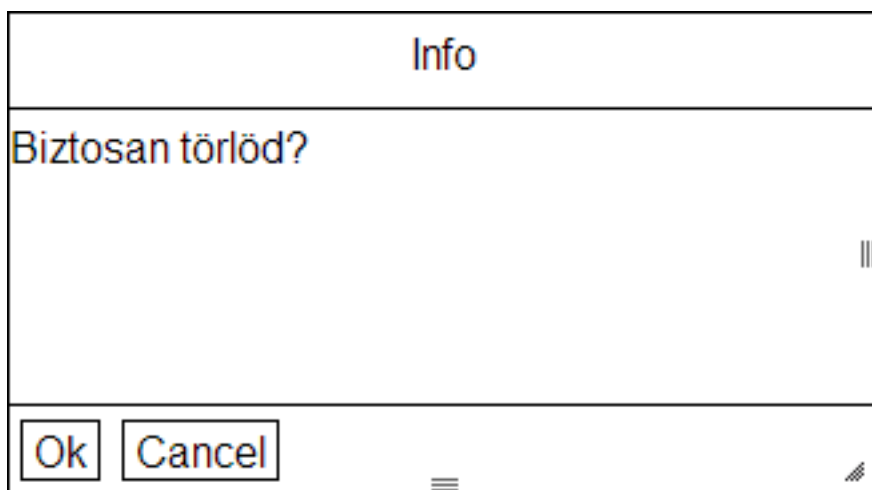
```
alert( 'Üzenet.' );
```

helyett

```
$( '#hv_alert_object' ).hv_alert( 'show_alert', {  
    'message': 'Üzenet.'  
});
```

a meghívás módja.

Ennek segítségével a weboldal egy CSS-el formázható ablakot jelenít meg, amely átméretezhető, áthelyezhető is lehet.



#### ***4.1.2. Felkérések***

A felkéréseknél kapunk tájékoztatást ismerős bejelölésről, vagy játékba való felkérésről.

A menüpontra belül fogadhatjuk el vagy tagadhatjuk meg ezeket a felkéréseket. Ezek ellenőrzése automatikus, a háttérben történik AJAX lekéréseken keresztül.

#### ***4.1.3. Ismerősök***

Ezen az oldalon listázódnak ki a már visszaigazolt ismerőseink, illetve újakat is itt kereshetünk. (A keresés névre keres.)

A már meglévő kapcsolatok törlésére is itt van lehetőség, minden akció Ajax lekéréseken keresztül zajlik.

Az ismerőseink bejelölése azért fontos, mert őket hívhatjuk majd be az elkészített játékaikba.

#### ***4.1.4. Profil módosítása***

Itt változtathatóak meg a profilbeállítások. Ezek a változtatási lehetőségek az e-mail címre, születési dátumra, nemre és jelszóra korlátozódnak.

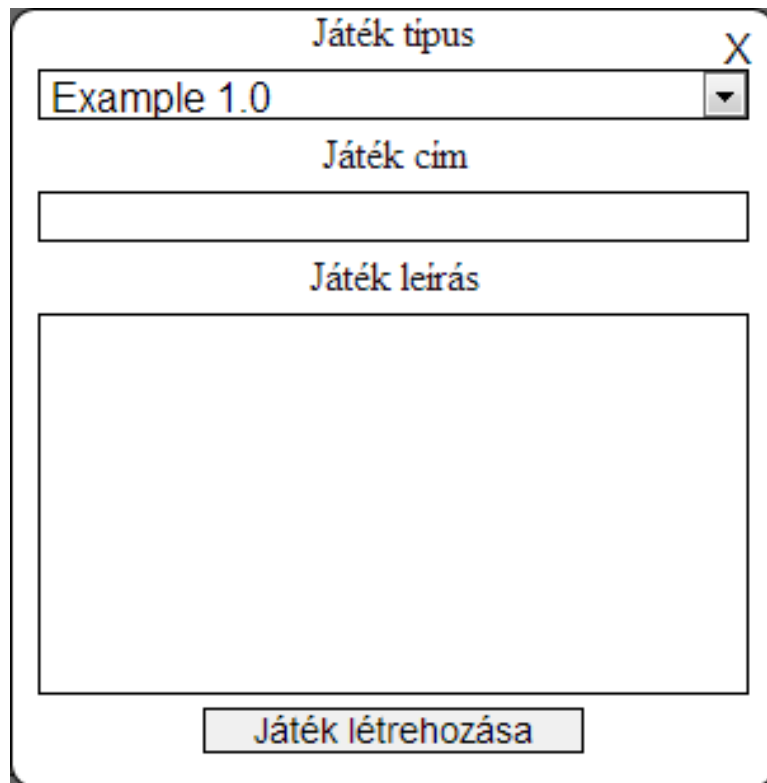
A név megváltoztatására nincs lehetőség.

#### ***4.1.5. Játékok kezelése***

Ezen a menüpontra belül jelennek meg a karaktermesterként, vagy résztvevőként játszott játékok.

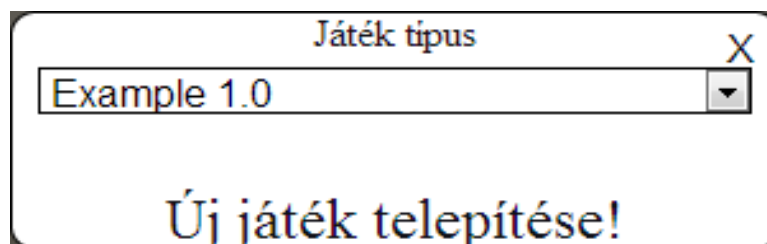
## Két menüpont érhető el

- Új játék indítása,



A screenshot of a graphical user interface window titled "Játék típus" (Game type) with a close button (X) in the top right corner. The window contains three input fields: a dropdown menu showing "Example 1.0", a text box for "Játék cím" (Game title), and a larger text area for "Játék leírás" (Game description). At the bottom center, there is a button labeled "Játék létrehozása" (Create game).

- Új játék telepítése.



A screenshot of a graphical user interface window titled "Játék típus" (Game type) with a close button (X) in the top right corner. The window contains a dropdown menu showing "Example 1.0" and a large text area displaying the text "Új játék telepítése!" (Install new game!).

Az első értelem szerűen egy új játékot indít a kiválasztott karakterlap típussal, melyben mi leszünk a Kalandmesterek. Itt lehetőségünk van kiválasztani a játéktípust az elérhető karakterlapokból, elnevezni a kalandot, és egy leírást adni hozzá. A kalandmester a játékon belül további adminfunkciókat ér el. (Utóbbit egy későbbi fejezetben tárgyalom.)

*Az elérhető játék típusok* ( Lényegében különböző karakterlapok. ) egy-egy adott felhasználóra vonatkozóan mást jelenthetnek.

Elérhető minden, a rendszer által alapértelmezetten támogatott játék. Ezek az összes

felhasználónál megjelennek.

Másrészt mindenkinek lehetősége van további, a számára tetsző karakterlapok telepítésére. Ezek mind olyan karakterlapok, amelyeket valaki elkészített, és átment az automatikus és manuális szűrőkön/felülvizsgálatokon.

Az itt megtalálható 'Új játék telepítése' gombon keresztül van lehetőségünk ilyen telepítésekre.

(Megjegyzés: Ha a kalandmester egy olyan játékot készít, amilyen karakterlap a játékosoknál nincs telepítve, nem okoz problémát. Csak a játék megalkotójánál kell, hogy a karakterlap telepítve legyen.)

A menüpontok alatt csatlakozhatunk a játékokhoz.

Ha a játékban éppen a kalandmester is jelen van, akkor itt zöld színnel ki lesz emelve. (Ennek ellenőrzése ismételten AJAX lekéréssel történik, mindaddig amíg a Játékok oldalon belül vagyunk.)

#### ***4.1.5. Fejlesztők (Karakterlap Fejlesztő Eszköz)***

A Fejlesztők menüponton belül további három menüpont fogad minket:

**1. Dokumentáció v0.7**

Az új dokumentáció.

**2. Régi dokumentáció v1.04.10 (~v0.3)**

A régi dokumentáció. (A karakterlapok rendszere átesett egy nagyobb átdolgozáson, így a régi és új dokumentáció sokban eltérnek.)

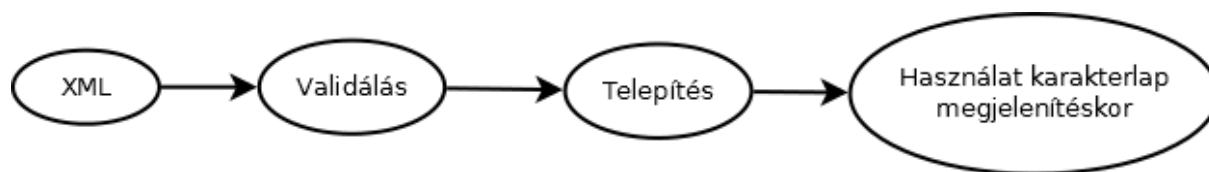
**3. Karakterlap Fejlesztő Eszköz.**

Erről az eszközről lesz bővebb szó.

A legújabb dokumentáció, és egyben a karakterlap verziószáma a 0.7-et kapta. Bár a karakterlap tökéletesen használható, további fejlesztési lehetőségek miatt kapta meg az 1.0 jelzöt. (A további fejlesztési lehetőségekről egy későbbi fejezetben lesz szó.)

A fejlesztő eszköz azért íródott, hogy megkönnyítse az új karakterlapok felvitelét.

Egy karakterlap elkészítése, majd használata a következő lépésekből áll:



Az ábrának megfelelően, először a karakterlap struktúráját leíró XML-t kell megírni. (Ennek szabályaival a dokumentáció foglalkozik.)

Miután ez elkészült, a Karakterlap Fejlesztő Eszközön keresztül ellenőrizhetjük le munkánkat:

**RPGCraft: Charsheet Developer Tool**

With this tool, you can check the character sheet written based on the documentation.  
Only the XML file is needed, the rest is optional. The tool will check it's validity, and if it seems good, a pop-up window will show you how the sheet will appear ingame.

XML file:  No file chosen  
Config file:  No file chosen  
Language INI file:  No file chosen or create this INI for me: ☐

-----Install mode-----  
Short name  Full name   
Install mode ☐ Forced install ☐

Itt feltölthetünk minden, a karakterlappal kapcsolatos fájlt, és amennyiben az XML megfelel az előírtaknak, megnézhetjük milyen lenne az játék közben.

(Itt a dokumentáció mellé is csatolt Example 1.0 nevű példajátékot mutatom be.)

A képen szereplő fehér téglalapon keresztül kapunk információt az ellenőrzésről.

Siker esetén egy ehhez hasonló szöveget kapunk:

```
Validity check on uploaded character sheet...
```

```
Checking 'base' tag... success.
```

```
Checking 'inventory' tag... success.
```

```
No errors.
```

```
No warnings.
```

```
Congratulations! Your character sheet passed the validation!  
Note that there could be errors at the installation process,  
this is no guarantee, just the first step.
```

A megjelenő pop-up ablak pedig:

Base		<input type="checkbox"/> Name name	<input type="checkbox"/> XP xp	<input type="checkbox"/> HP hit_points	<input type="checkbox"/> Defence defense	<input type="checkbox"/> Damage damage
Inventory						
<input checked="" type="checkbox"/>	Name name	<input checked="" type="checkbox"/> Quantity quantity	up			
<input type="checkbox"/>	Name name	<input type="checkbox"/> Quantity quantity	del			

Három fájl tölthető fel:

- XML fájl (kötelező),
- Konfigurációs ini fájl (nem kötelező),
- Nyelvi ini fájl (nem kötelező).

A karakterlap működéséhez minden fájl tartalmára különböző szabályok érvényesek.

A nyelvi fájlt generáltathatjuk is az eszközön keresztül, így nem kell fejben tartanunk az elnevezési konvenciókat, elég ha átírjuk a generált fájl tartalmát.



Az Example 1.0 karakterlap fájljai:

<code>example_10</code>	<code>xml</code>
<code>lang_example_10_en</code>	<code>ini</code>
<code>lang_example_10_hu</code>	<code>ini</code>

Láthatóan a karakterlap több nyelvet is támogat. A rendszer a karakterlap betöltésekor először megpróbálja betölteni az oldalon, a felhasználó által beállított nyelvet, amennyiben pedig ez nem áll rendelkezésre, az alapértelmezettet. Ha az alapértelmezett nyelvi fájl sem létezik (Ami csak a karakterlap telepítésekor elkövetett hanyagságból adódhat.), akkor a karakterlap címkék nélkül, üresen töltődik be.)

Így például ha egy karakterlaphoz csak angol nyelvi fájl adott, és az oldalon a felhasználó a német nyelvet állítaná be (Ami ebben a verzióban nem támogatott.), a következőképpen töltődne be a karakterlap:

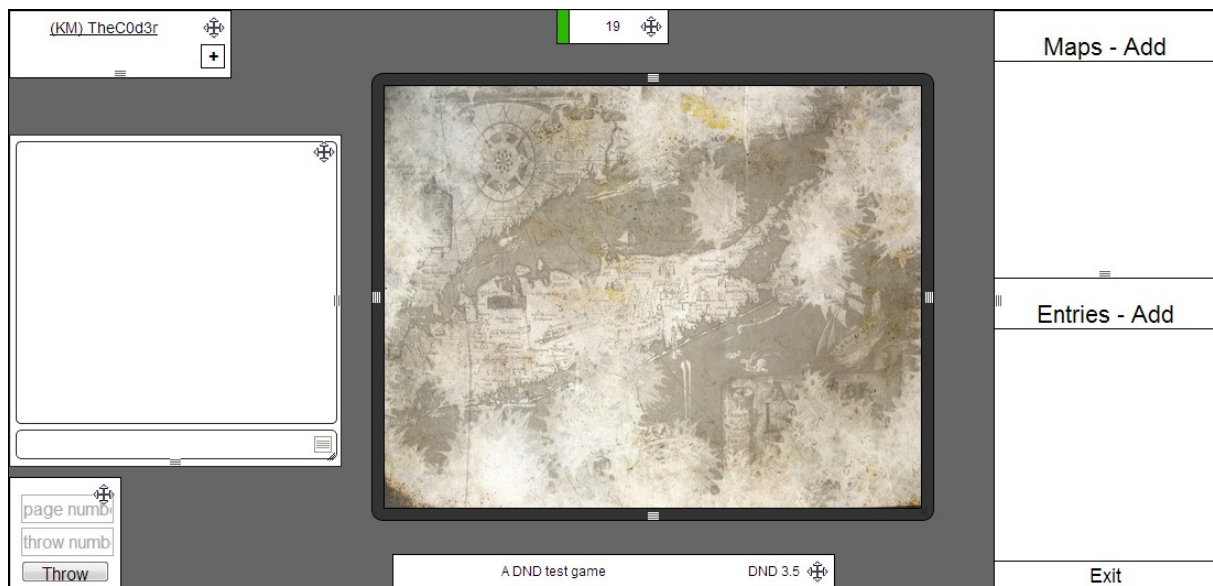
Először keresne német nyelvet, majd mivel nem létezik, megpróbálná az alapértelmezett magyart betölteni. Mivel ez sem áll rendelkezésre, a német felhasználónak végül angolul jelenne meg a karakterlap.

A *karakterlap telepítésére* az admin jogosultsággal rendelkező felhasználó képes. Számára a Karakterlap Fejlesztő Eszközben megjelenik egy telepítési szekció. Itt megadhatja a karakterlap rövid és teljes nevét (pl: `example_10`, `Example v1.0`), majd a csatolt fájlok alapján telepítheti a karakterlapot. Ehhez az `'install mode'`-ot kell kipipálni, illetve a `'forced mode'` kipipálásával a figyelmeztetéseket (warningokat) és el tudjuk nyomni, azok figyelmen kívül hagyásával települ a karakterlap.

Sikeres telepítés után minden felhasználó elérheti a karakterlapot telepítésre.

(Karakterlapot telepítés nélkül is elérhetővé a *supported\_games\_list.ini* fájl módosításával lehet tenni.)

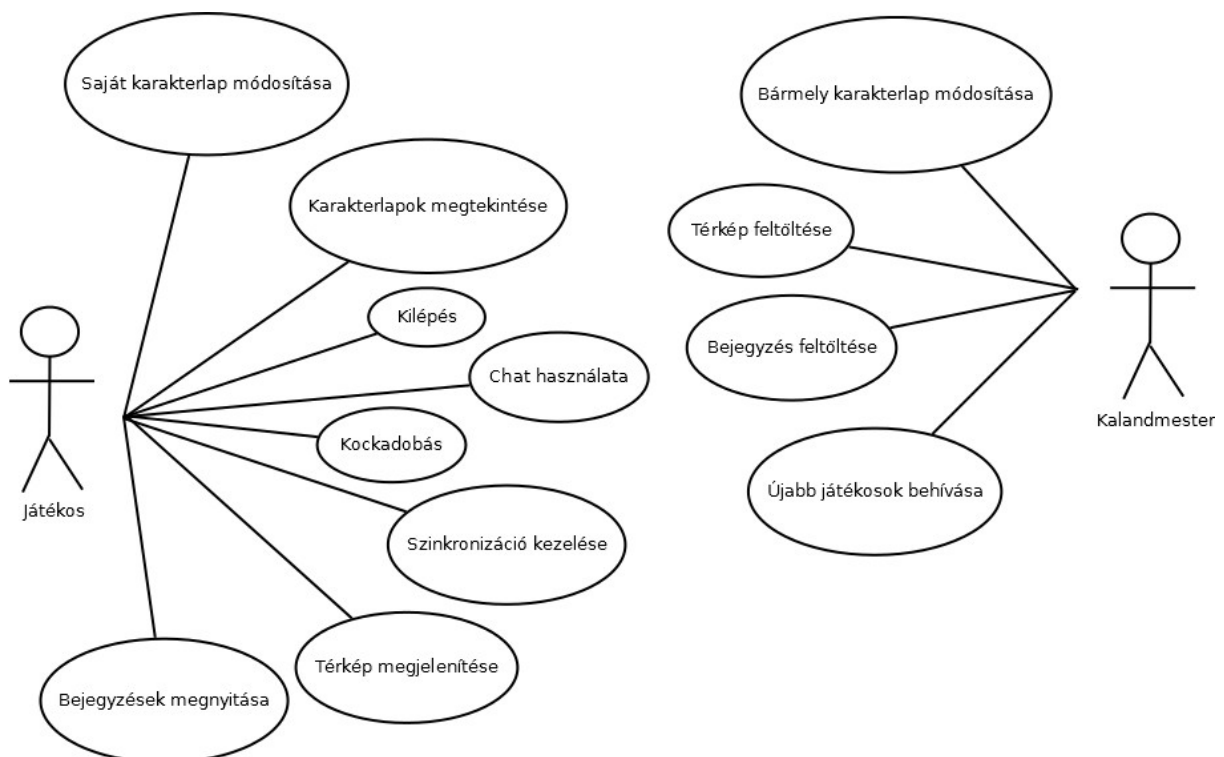
## 4.2. Játék színtér



A képen egy új, üres színtér látható.

A játék színtér a rendszer lelke. Itt lehet a tényleges játékot megkezdeni, használni a karakterlapot, beszélni a többi játékosal.

### Összesítés az elérhető funkciókról



A Kalandmester a játékos funkciókat is eléri, viszont saját karakterlapja nincsen.

#### 4.2.1. Ajax lekérések

A játék interface nagyban épül az úgynevezett AJAX (Asynchronous JavaScript and XML) lekérésekre. Ezek segítségével az oldal újratöltése nélkül, a háttérben folyhatnak XHR-ek (XMLHttpRequest).

Az elnevezés ebben az esetben félrevezető lehet, mert XML helyett JSON-ben (JavaScript Object Notation) formázva használom a szükséges adatokat.



A képen látható a rendszerben megvalósított AJAX lekérések általános formája.

A lekérés a háttérben fut le a megfelelő paramétereket egy PHP fájlra átadva.

Miután a PHP a kért adatokat összegyűjtötte, egy JSON állománnyal tér vissza.

#### Példa JSON-re

```
{
  "in_game_users": [
    {
      "userID": "1",
      "time": "1368465692.7231",
      "name": "admin",
      "active": "false"
    },
    {
      "userID": "61",
      "time": "1368474819.2133",
      "name": "asd",
      "active": "true"
    }
  ]
}
```

Ezt végül az adatokat kérő JavaScript funkció dolgozza fel.

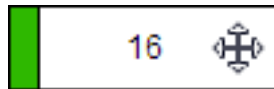
#### 4.2.2. Szinkronizálás

A játék színterén több olyan elem található, amely időnként frissülhet:

- új játékosok csatlakozása,
- játékosok aktívvá/inaktívvá válása,
- térkép hozzáadása/törlése,
- bejegyzés hozzáadása/törlése,
- új hozzászólás/kockadobás chat ablakban.

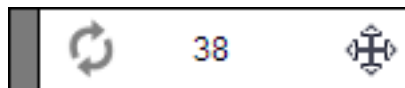
Mindezek frissítésére egy szinkronizáló AJAX lekérés szolgál. Ez bizonyos időközönként meghívódik, és egy JSON objektumot ad vissza a színtéren történő változásokról.

A szinkronizációk pontos követésére egy újabb elem szolgál az interfészen:



Az itt látható szám a lekérés teljesítését mutatja meg miliszekundumban. (1000ms=1s)

Ha a bal szélén elhelyezett téglalap zöld, a szinkronizáció automatikusan történik. Lehetőség van viszont átváltani manuális módra. Ennek oka lehet például a lassú internetkapcsolat, túlterhelés. Ha 1-2 másodperc (1000-2000ms) körül van a frissítési idő, érdemes manuális módra kapcsolni a zöld téglalapra való kattintással:



Ettől fogva a szinkronizáció csak a megjelent frissítő ikonra kattintva fog megtörténni. Az itt szereplő szám a legutóbbi lekérésre vonatkozik.

Újra a téglalapra kattintva vissza válthatunk automatikus módra.

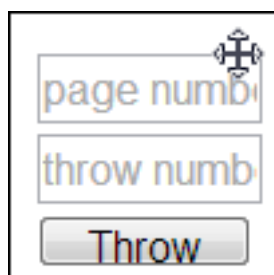
Minden szinkronizációs alkalommal egyúttal magunkról is „életjelet” adunk a többi játékos számára, ez alapján dől el, hogy valaki aktív, vagy inaktív-e egy játékban.

Így ha manuálisan frissítjük a színteret, a többi játékos sokszor inaktívnak fog látni a színtéren belül.

#### 4.2.3. Chat

A chat független a szinkronizációtól, mindig aktív. Itt láthatóak a játékosok üzenetei, és a kockadobások is.

#### 4.2.4. Kockadobás

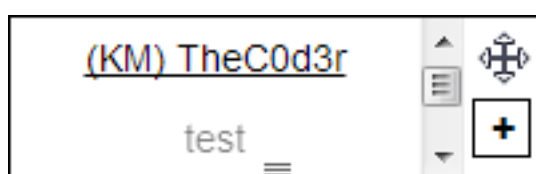


Ezen a felületen adható meg, hogy hány oldalú kockával mennyiszer szeretnénk dobni.

Az eredmény megjelenik a chat-ablakban üzenetként, így azt a többi játékos is ellenőrizheti.

Fontos itt megemlíteni, hogy az itt szereplő „véletlen” szám nem egészen véletlen, csak úgynevezett pseudo-random szám. Ennek orvoslására fejlesztésképpen lehetne használni valódi véletlen (true random) szám generátorokat (pl.: random.org API).

#### 4.2.5. Csatlakozott játékosok



Ezen a felületen láthatóak a játékba csatlakozott játékosok, elsőként a Kalandmester. Utóbbi neve elé mindig egy „(KM) ” jelző kerül.

A játékosok lehetnek aktívak, vagy inaktívak, a szinkronizációtól függően. Ha egy játékos 10 másodpercnél régebb óta nem szinkronizált, a rendszer 10-től visszaszámol, majd pedig inaktívvá teszi, a nevét elhalványítja.

A játékosok nevére kattintva megtekinthető a karakterlapuk. Kétféle módon történhet itt a karakterlap megjelenítése:

- nézőként, aki nem módosíthat semmit (Csak a láthatóvá állított elemeket látja.),
- tulajdonosként/Kalandmesterként, aki módosíthat bármit.

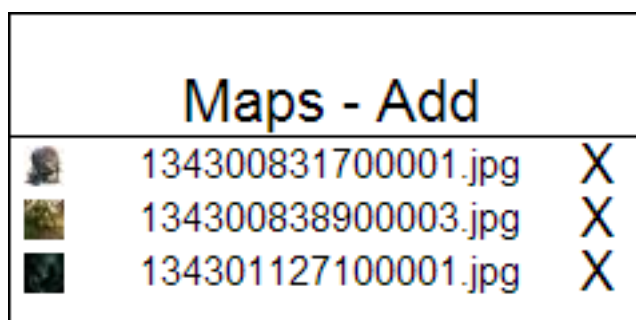
Az itt szereplő „+” gombbal hívhat be a Kalandmester új játékosokat az ismerősei közül.



A megjelenő ablak bal oldalán szerepelnek a behívható ismerősök (akik már visszaigazoltak), a jobb oldalon pedig a behívottak szerepelnek.

A behívottaknak a weboldalon a felkérések résznél kapnak értesítést a játékba invitálásról. Amennyiben elfogadják, a játékhoz lesznek átirányítva.

#### 4.2.6. Térképek



A szintér jobb oldalán szerepelnek a hozzáadott térképek. (Alapvetően térképek megosztására van kitalálva ez a rész, de a Kalandmester döntése szerint bármit felvihet ide.) A Kalandmester számára a képen látható módon lehetősége van a térképek törlésére,

hozzáadására. Hozzáadáskor több képet is kiválaszthatunk amelyek a feltöltés végeztével megjelennek a lista végénél.



Ha egy térképre rákattintunk, akkor a háttérben lévő képnézegető a kiválasztott képet tölti be. A betöltött képet kedvünk szerint helyezhetjük át a képernyőn, illetve a széleinél méretezhetjük.

Ez a kép jelenik meg, ha nincs feltöltött térkép.

#### ***4.2.7. Bejegyzések***

**Entries - Add**  
**Khelben Arunsun X**

Úgyszint a jobb oldalon, a térképek listája alatt a bejegyzések menüje található. Ennek kezelése megegyezik az előzővel.

Hozzáadáskor a megjelenő ablakban megadhatjuk a bejegyzés címét, a hozzá tartozó



szöveget, illetve beállíthatjuk, hogy a játékosok közül kiknek legyen látható.

Ha a létrehozást követően megnyitjuk, képek csatolására is lehetőségünk van.

The screenshot shows a game window titled "Khelben Arunsun" with standard window controls (Up, X, and a move icon). The main text area contains a description of Khelben Blackstaff Arunsun, a Lord Mage of Waterdeep, and a scroll bar on the right. Below the text is a "Visibility" section with radio buttons for "All", "None", and "Other:" (which is selected). Under "Other:", there are checkboxes for "test" (checked) and "control". A horizontal scroll bar is positioned below these options. At the bottom left, there is a small icon of a wizard. At the bottom center, there is a button labeled "Upload pic".

**Khelben Arunsun** Up X

Khelben Blackstaff Arunsun (414 DR – 1374 DR, aged 960 years) was the Lord Mage of Waterdeep, a masked Lord of Waterdeep, and powerful archmage.[citation needed] He resided in Blackstaff Tower until his death.  
[1]

One of the few spellcasters to become one of Mystras Chosen,[2][3] and last survivor of House Maerdrym,[4] Khelben was a serious, imposing wizard, and a grim proponent of law and order whose methods of achieving the ultimate good were often highly questionable in the eyes of the righteous. One such deal—with Fzoul Chembryl, cleric of Bane and leader of the sinister Zhentarim—led the once highly regarded member of the semi-secret Harper organization to simultaneously be expelled from and leave the organization. Khelben did not leave alone, however; he formed the Harper splinter group the Moonstars to better suit his needs.

**Visibility**

☐ All ☐ None ☒ Other:

☒ test ☐ control

Upload pic

Ha ebben az ablakban módosítjuk a bejegyzést, az „Up” gombra kattintással véglegesíthetjük azt. (Ezt követően a változás a többi játékos számára is látható lesz.)



### 4.2.8. Karakterlap

A rendszer legfontosabb része, létrehozását a korábbi fejezetekben fejtettem ki.

Az itt megjelenő karakterlap teljes mértékben az azt leíró XML-től függ. Ez alapján a rendszer felépíti a képen látható beviteli mezőket, AJAX lekéréseket.

#### Tulajdonos, vagy Kalandmester által megnyitott karakterlap

A mezők tartalma és láthatósága módosítható.

Alap	<input checked="" type="checkbox"/> name	<input checked="" type="checkbox"/> level	<input checked="" type="checkbox"/> xp	<input checked="" type="checkbox"/> race	<input checked="" type="checkbox"/> class	<input checked="" type="checkbox"/> alignment	<input checked="" type="checkbox"/> deity	<input checked="" type="checkbox"/> size	X
	Wilben Darksb	12	70680	Moon Elf	Rogue	Natural Good	Seldarine, Selu	medium	
	<input checked="" type="checkbox"/> age	<input checked="" type="checkbox"/> gender	<input checked="" type="checkbox"/> height	<input checked="" type="checkbox"/> weight	<input checked="" type="checkbox"/> eyes	<input checked="" type="checkbox"/> hair	<input checked="" type="checkbox"/> skin	<input checked="" type="checkbox"/> speed	
	128	male	165cm	70kg	zöld	fehér	kék	30	
	<input checked="" type="checkbox"/> dmgRed								
	0								
HP	<input checked="" type="checkbox"/> hpTotal	<input checked="" type="checkbox"/> poundsCurrent	<input checked="" type="checkbox"/> nonlethal						
	94	15/94	0						
Pénz	<input checked="" type="checkbox"/> moneyCP	<input checked="" type="checkbox"/> moneySP	<input checked="" type="checkbox"/> moneyGP	<input checked="" type="checkbox"/> moneyPP					
	63	8	4	0					
Páncél	<input checked="" type="checkbox"/> gearArName	<input checked="" type="checkbox"/> gearArType	<input checked="" type="checkbox"/> gearArAcBonus	<input checked="" type="checkbox"/> gearArMaxDex	<input checked="" type="checkbox"/> gearArCheckPen	<input checked="" type="checkbox"/> gearArSpellFail	<input checked="" type="checkbox"/> gearArSpeed	<input checked="" type="checkbox"/> gearArWeight	
	Elven chain	-	0	2	0	0	0	0	
	<input checked="" type="checkbox"/> gearArSpecProp								
	0								
Pajzs	<input checked="" type="checkbox"/> gearShName	<input checked="" type="checkbox"/> gearShAcBonus	<input checked="" type="checkbox"/> gearShWeight	<input checked="" type="checkbox"/> gearShCheckPen	<input checked="" type="checkbox"/> gearShSpellFail	<input checked="" type="checkbox"/> gearShSpecProp			
	-	0	0	0	0	0			
Védő tárgy	<input checked="" type="checkbox"/> prot1Name	<input checked="" type="checkbox"/> prot1AcBonus	<input checked="" type="checkbox"/> prot1Weight	<input checked="" type="checkbox"/> prot1SpecProp					
	Ring of Protecti	3	0	-					
Védő tárgy	<input checked="" type="checkbox"/> prot2Name	<input checked="" type="checkbox"/> prot2AcBonus	<input checked="" type="checkbox"/> prot2Weight	<input checked="" type="checkbox"/> prot2SpecProp					
	-	0	0	-					
AC	<input checked="" type="checkbox"/> acTotal	<input checked="" type="checkbox"/> stat_10	<input checked="" type="checkbox"/> ArmorBonus	<input checked="" type="checkbox"/> ShieldBonus	<input checked="" type="checkbox"/> acDexMod	<input checked="" type="checkbox"/> acSizeMod	<input checked="" type="checkbox"/> NaturalArmor	<input checked="" type="checkbox"/> acDefMod	
	23(24) 21/33	10 +	6(7)	0	4	0	0	3	
	<input checked="" type="checkbox"/> acMiscMod	<input checked="" type="checkbox"/> mage Reductio							
	0	0							
STR	<input checked="" type="checkbox"/> strAbScore	<input checked="" type="checkbox"/> strAbMod	<input checked="" type="checkbox"/> strTempScore	<input checked="" type="checkbox"/> strTempMod	DEX	<input checked="" type="checkbox"/> dexAbScore	<input checked="" type="checkbox"/> dexAbMod	<input checked="" type="checkbox"/> dexTempScore	<input checked="" type="checkbox"/> dexTempMod
	16	3	0	0		4	0		
	<input checked="" type="checkbox"/> dexAbScore	<input checked="" type="checkbox"/> dexAbMod	<input checked="" type="checkbox"/> dexTempScore	<input checked="" type="checkbox"/> dexTempMod		<input checked="" type="checkbox"/> touch			

## Más játékos által megnyitott karakterlap

<b>Alap</b>	name	level	xp	race	class	alignment	deity	size
	Wilben Darksb	12	70680	Moon Elf	Rogue	Natural Good	Seldarine, Selu	medium
	age	gender	height	weight	eyes	hair	skin	speed
	128	male	165cm	70kg	zöld	fehér	kék	30
	dmgRed							
	0							
<b>HP</b>	hpTotal	woundsCurrent	nonlethal					
	94	15/94	0					
<b>Pénz</b>	moneyCP	moneySP	moneyGP	moneyPP				
	63	8	4	0				
<b>Páncél</b>	gearArName	gearArType	gearArAcBonus	gearArMaxDex	gearArCheckPen	gearArSpellFail	gearArSpeed	gearArWeight
	Elven chain	-	0	2	0	0	0	0
	gearArSpecProp							
	0							
<b>Pajzs</b>	gearShName	gearShAcBonus	gearShWeight	gearShCheckPen	gearShSpellFail	gearShSpecProp		
	-	0	0	0	0	0		
<b>Védő tárgy</b>	prot1Name	prot1AcBonus	prot1Weight	prot1SpecProp				
	Ring of Protecti	3	0	-				
<b>Védő tárgy</b>	prot2Name	prot2AcBonus	prot2Weight	prot2SpecProp				
	-	0	0	-				
<b>AC</b>	acTotal	stat_10	acArmorBonus	acShieldBonus	acDexMod	acSizeMod	acNaturalArmor	acDefMod
	23(24) 21/33	10 +	6(7)	0	4	0	0	3
	acMiscMod	Damage Reductio						
	0	0						
<b>STR</b>	strAbScore	strAbMod	strTempScore	strTempMod	<b>DEX</b>	iniDexMod	iniMiscMod	
	16	3	0	0		4	0	
	dexAbScore	dexAbMod	dexTempScore	dexTempMod		touch		

Egy adott személy az aktuális játékhoz tartozó karakterlapjának betöltése a következő lépésekben történik:

Karakterlap előtöltés → Struktúra felépítése → Adatokkal való kitöltés

A játék indításakor a rendszer a karakterlapot is előtölti. Ennek során feldolgozza a tárolt XML fájlt, átalakítja JSON formátumba, majd eltárolja azt későbbi használatra.

Ennek előnye, hogy nem kell minden karakterlap megnyitásakor újra lekérni, majd bejárni az XML fájlt, elég ha a letárolt struktúrát a könnyebben bejárható JSON-ból vesszük elő. Így csak az adatbáziskapcsolatra kell várni, hogy az a karakterlapot feltöltse adatokkal.

Ha Kalandmesterként, vagy tulajdonosként nyitjuk meg a karakterlapot, itt lehetőségünk van azt kitölteni/módosítani. Kitöltéskor nincs más dolgunk, mint beírunk a megfelelő adatokat. Ezt követően, ha tíz másodpercig nem történik további módosítás, a rendszer automatikusan felviszi a változtatásokat, nincs szükség külön gomb lenyomásra.

Az elemek kitöltése mellett megadhatjuk azok láthatóságát is más játékosok számára.

## 5. Karakterlap dokumentáció

Az itt szereplő dokumentáció csak a ténylegesen elkészült funkcionalitást mutatja be, míg a mellékletként csatolt verzióban a tervezett funkciók is szerepelnek.

### Verziószám

A verzió számozása megváltozott. Az előző szám 1.04.10 volt, ami a dátumból készült. Hogy követhetőbb legyen a fejlesztések mértéke is, hagyományosabb számozásra térek.

Az aktuális verziószám így 0.7 lett, míg az előzőt ~0.3-ra lehet becsülni.

### XML-struktúra

A teljes karakterlaphoz a következő fájlok kellenek:

Struktúra XML-fájl: `<név>.xml` (kötelező)

Nyelvi ini fájl(ok): `lang_<név>_<nyelv>.ini` (legalább egy nyelv)

Configurációs ini fájl: `conf_<név>.ini` (opcionális)

### Példa

`example_10.xml`

`lang_example_10_hu.ini`

`lang_example_10_en.ini`

`conf_example_10.ini`

Ezek a fájlok csatolva lettek a dokumentációhoz.

## 5.1. *Magyarázat a kódokhoz*

### **example\_10.xml**

Az alapvető struktúra:

```
<?xml version="1.0" encoding="utf-8"?>
<charsheet>
  <maintag limit="1">
    <record>
      <field name="fieldname" />
      <field name="fieldname" />
    </record>
    <record>
      <field name="fieldname" />
      <field name="fieldname" />
    </record>
  </maintag>
</charsheet>
```

A charsheet tag, a hozzá tartozó limit attribútum és az első sor kötelező. Ezen belül tetszőleges számú maintag, record és field szerepelhet ebben az egymásba ágyazási sorrendben. A field name attribútum szintén kötelező, ez alapján generálódik a nyelvi fájl.

(Technikai okok miatt minden field névnek különböznie kell.)

A többi attribútum a dokumentáció végén szerepel.

A main tagek 'limit' attribútuma határozza meg, hogy a tag egyszer, vagy többször is felvihető-e.

Példa:

Az alapvető értékek, mint: health, damage csak egyszer szerepelnek, így limit="1"-es taggel szerepelnek.

Ha tárgyakat akarunk felvinni, limit="0" taget használjunk, így tudja a rendszer, hogy ebből akárhány példány felvihető.

(Jelenleg a 0/1 értékek támogatottak.)

## Nyelvi fájlok

A nyelvi fájl letölthető az oldalon a Karakterlap Fejlesztővel. Az elnevezésre kell csak figyelni.

Az 'author' és 'desc' elemek kiemelve, ezek jelennek meg in-game a karakterlap részleteinél. (Ezért a field tag-ek nevéként sem szerepelhet ez a kettő.)

### conf\_example\_10.ini

Egy config fájl kell csak a karakterlaphoz.

Az itt szereplő beállítások közül nem mindegyik támogatott még.

(Halványabban vannak megjelenítve.)

charsheet_noresize	false/true. Ha true, a karakterlap nem átméretezhető ingame.
resize_min_width	Az átméretezés min. szélessége.
resize_max_width	Az átméretezés max. szélessége.
resize_min_height	Az átméretezés min. magassága.
resize_max_height	Az átméretezés max. magassága.

## 5.2. XML dokumentáció

### main tag

- limit
  - kötelező
  - lehet: 0/1
  - 0: több felvihető sort eredményez, a „végtelen” szinonímája.
  - 1: csak egy, fix példány az adott tagekből.
- left\_title, right\_title, top\_title, bot\_title
  - opcionális
  - A feltöltő tag címe.  
Ekkor a rekordok közösen kapnak egy címet, nem pedig rekordonként egyedit.
- lower\_label\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címkék mindegyikére vonatkozni fog.
- lower\_input\_css
  - opcionális
  - Az itt megadott css kód a leszármazott inputok mindegyikére vonatkozni fog.
- lower\_title\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címek mindegyikére vonatkozni fog.
- lower\_label\_hover\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címkék mindegyikére vonatkozni fog, ha az egér felette van.
- lower\_input\_hover\_css
  - opcionális
  - Az itt megadott css kód a leszármazott inputok mindegyikére vonatkozni fog, ha az egér felette van.
- lower\_title\_hover\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címek mindegyikére vonatkozni fog, ha az egér felette van.
- title\_css
  - opcionális
  - Csak ha 0-limites tagról van szó, a cím css-kódját itt lehet megadni, nem a record title\_css attribútumánál.
- css
  - opcionális
  - Az aktuális elemre vonatkozó css kód adható meg.
- hover\_css
  - opcionális
  - Az aktuális elemre vonatkozó css kód adható meg, ha az egér felette van.

## **record**

- left\_title, right\_title, top\_title, bot\_title
  - opcionális
  - A sor egyedi címet kap.
- lower\_label\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címkék mindegyikére vonatkozni fog.
- lower\_input\_css
  - opcionális
  - Az itt megadott css kód a leszármazott inputok mindegyikére vonatkozni fog.
- lower\_label\_hover\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címkék mindegyikére vonatkozni fog, ha az egér felette van.
- lower\_input\_hover\_css
  - opcionális
  - Az itt megadott css kód a leszármazott inputok mindegyikére vonatkozni fog, ha az egér felette van.
- title\_css
  - opcionális
  - Az itt megadott css kód a leszármazott címek mindegyikére vonatkozni fog.
- css
  - opcionális
  - Az aktuális elemre vonatkozó css kód adható meg.
- hover\_css
  - opcionális
  - Az aktuális elemre vonatkozó css kód adható meg, ha az egér felette van.

## **field**

- name
  - kötelező
  - Ez lesz az adat egyedi elnevezése. (Egyedi kell legyen!)
  - Amennyiben nem adunk meg placeholder attribútumot, úgy ez a név jelenik meg üres input-dobozban is.
- placeholder
  - opcionális
  - Az aktuális mező beviteli mezőjében jelenik meg, ha az üres.
- css
  - opcionális
  - Az aktuális elemre vonatkozó css kód adható meg.
- label\_css
  - opcionális
  - Az itt megadott css kód a címkére vonatkozik.
- input\_css
  - opcionális
  - Az itt megadott css kód az inputra vonatkozik.
- hover\_css

- opcionális
  - Az aktuális elemre vonatkozó css kód adható meg, ha az egér felette van.
- label\_hover\_css
  - opcionális
  - Az itt megadott css kód a címkére vonatkozik, ha az egér felette van.
- input\_hover\_css
  - opcionális
  - Az itt megadott css kód az inputra vonatkozik, ha az egér felette van.
- visible
  - opcionális
  - alapértelmezett: true
  - értékei lehetnek: true/false
  - Az adott mező default láthatóságát állíthatjuk be.
- switchable
  - opcionális
  - alapértelmezett: true
  - értékei lehetnek: true/false
  - Beállítja, hogy a user változtathatja-e a default láthatóságot.
- static\_content
  - opcionális
  - alapértelmezett: null
  - Ha megadjuk ezt az attribútumot, úgy az nem fog kitölthető inputként szerepelni a karakterlapon, de a láthatósága beállítható marad. Értékként a megadott szöveget fogja kapni input-mező helyett, fixen.
- default
  - opcionális
  - alapértelmezett: null
  - Megadja a mező alapértelmezett értékét. Ez akkor jelenik meg, ha az adott mező kitöltése nem kötelező, és feltöltéskor üresen hagyják.



## Összefoglaló

A feladatkiírásban szereplő funkciók mind elkészültek a rendszerben, illetve azon túl is mutat. Ilyen többek között a karakterlap is.

A rendszer a fejlesztés során több alkalommal újra- , illetve átíródott, valahányszor optimalizálási lehetőségre leltem, vagy fejlődtem egy-egy technológia használatában.

Ilyen volt a karakterlap fejlesztése is, amely eleinte csak fix, manuálisan létrehozott karakterlapokkal működött, melyekhez külön létre kellett hozni a felvitelhez szükséges űrlapokat és adattáblákat. E módszer hátrányait felismerve született meg az XML-el könnyedén létrehozható karakterlapok nyelve, és az azt feldolgozó értelmező PHP kód.

Ekkor a v0.3-mas verziószámot kapta meg, majd további fejlesztések során jött létre a jelenlegi, v0.7-es verzió.

Az elnevezés nem véletlenszerű, ezzel szerettem volna érzékelteni, hogy még mennyi mindent el lehetett volna készíteni az 1.0. verzióig.

Ehhez hasonló fejlődésen esett át a játék szintér AJAX szinkronizálása is, amely fejlesztés lépcsőfokait felsorolni meg se kísérek.

Sok további bővítésre is volna lehetőség, így például:

- A karakterlap tovább fejlesztése,
- Online WYSIWYG (What You See Is What You Got) szerkesztő készítése karakterlap XML struktúrák létrehozására,
- A térképek mezőkre bontása, elemek elhelyezése rajta.

A rendszer demóját meg lehet tekinteni az alábbi linken:

<http://HubertViktor.com/RPGCraft/>

## **Irodalomjegyzék**

- [1] <http://en.wikipedia.org/wiki/HTML>
- [2] [http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [3] <http://en.wikipedia.org/wiki/XML>
- [4] [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [5] [http://en.wikipedia.org/wiki/LESS\\_\(stylesheet\\_language\)](http://en.wikipedia.org/wiki/LESS_(stylesheet_language))
- [6] <http://en.wikipedia.org/wiki/PHP>
- [7] <http://en.wikipedia.org/wiki/MySQL>
- [8] <http://en.wikipedia.org/wiki/JavaScript>
- [9] <http://en.wikipedia.org/wiki/JQuery>

## Nyilatkozat

Alulírott Hubert Viktor Programtervező informatikus Bsc szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Tanszékcsoport Szoftverfejlesztés Tanszékén készítettem, Programtervező informatikus Bsc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem Informatikai Tanszékcsoport könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2013. Január 31.

.....  
Aláírás

## Melléklet

Mellékletben csatoltam egy cd-t, melyen a szakdolgozatom implementációja, adatbázis, a szakdolgozatom, és a beüzemeléssel kapcsolatos információk találhatóak. Emellett még a karakterlaphoz tartozó dokumentációk is itt találhatóak meg. A rendszer üzembe helyezéséhez a következők szükségesek:

- Apache szerver
- MySQL szerver (phpMyAdmin ajánlott)
- Böngésző (Ajánlott Chrome, vagy Firefox)

Az adatbázisban tesztfelhasználók szerepelnek, melyek véletlenszerűen kapcsolatban állnak egymással. Felkerült pár karakterlap és ezekkel indított játék is.

Részletes információk az „!infos” mappában találhatóak.

### Tartalom

- *Karakterlap dokumentációk:* 0.3 és 0.7-es verziók.
- *Szakdolgozat:* Jelen szakdolgozat több formátumban.
- *Telepítés:* Részletes leírás a rendszer telepítésével kapcsolatban, és az ehhez szükséges fájlok.

### Létrehozott felhasználók

- admin:  
A *Karakterlap Feltöltő Eszköz* telepítő része ezen a felhasználón keresztül érhető el..
  - e-mail: admin@admin.admin
  - jelszó: admin
- tester
  - e-mail: tester@tester.tester
  - jelszó: tester

A rendszer tesztelése egy gépen, több böngészőn keresztül is lehetséges.

## **Köszönetnyilvánítás**

Köszönöm a téma meghirdetését és a dolgozat elkészülése alatt nyújtott segítséget

Havasi Ferencnek és Török Tibornak

Köszönet továbbá mindazoknak akik a rendszert tesztelték, ötleteikkel gazdagították.