

GURPS hahmolomake

Tietokantasovellus 2012

JP Myllykangas

24. maaliskuuta 2012

1 Johdanto

1.1 Järjestelmän tarkoitus

Sovellus on verkossa toimiva interaktiivinen hahmolomake **GURPS**¹ roolipeli-järjestelmää varten. Pelissä hahmoilla on runsaasti erilaisia ominaisuuksia joista pelaajan täytyy pitää kirjaa. Osa ominaisuuksista johdetaan matemaattisesti toisista ja osa muuttuu usein, joten automaattinen laskenta helpottaa hahmojen ylläpitoa.

Perusidea GURPSin hahmoissa on laskea pistesummaa: hyödylliset ominaisuudet (advantages) ja taidot (skills) kasvattavat hahmon pistesummaa, kun heikkoudet (disadvantages & quirks) taas vähentävät kokonaispistesummaa. Täten käytännössä kaikilla hahmoilla on joukko vahvuuksia ja taitoja, jotka kasvattavat pistesummaa ja sen jälkeen joukko tasapainottavia heikkouksia jotka vähentävät niitä. Yleensä peleissä pyritään alussa noin 100 pisteen hahmoihin. Esimerkki valmiista hahmolomakkeesta löytyy kuvasta **1**.

Pelaajat voivat luoda ja ylläpitää useita hahmolomakkeita, joiden tiedot tallennetaan tietokantaan. Johdettuja ominaisuuksia kuten esim. nopeus² ei säilötä kantaan vaan ne lasketaan kun lomake tulostetaan ruudulle. Koska heikkoudet, vahvuudet ja taidot ovat kaikille hahmoille yhteisiä, voidaan ne säilöä omiin tauluihinsa tietokannassa.

Jokaisella pelaajalla on oma käyttäjätili, joka antaa heille oikeuden muokata omia hahmojaan ja tarkastella muiden hahmoja. Pelaajat voivat myös luoda uusia taitoja, vahvuuksia ja heikkouksia. Lisäksi sovelluksessa on ylläpitäjä joka voi luoda uusia käyttäjätiliä.

Sovellus tulee olemaan englanniksi, sillä kaikki lähdemateriaali on sillä kielellä.

1.2 Toimintaympäristö

Kyseessä on internetissä toimiva interaktiivinen html - sivusto. Sovellus toteutetaan pääosin PHP - ohjelmointikielellä, mutta osa lomakkeen kentistä päivitetään JavaScriptillä. Tietokantana käytetään MySQL:ää.

¹Generic Universal Roleplaying System

²Nopeus lasketaan summaamalla pelaajan ketteryys (DX) ja terveys(HT) - attribuutit ja jakamalla tulos neljällä

GURPS

CHARACTER SHEET

Name: DAI BLACKTHORN Player
 Appearance: 5'6", 110 lbs., FAIR SKIN, DARK HAIR and EYES
 Character Story: STREET KID, THIEF

Date Created: _____ Sequence: _____
 Unspent Points: _____ Point Total: **100**

Pl. Cost
ST 8
DX 15
IQ 12
HT 12

FATIGUE

BASIC DAMAGE

Thrust: 1d-3

Swing: 1d-2

HITS TAKEN

Mvmt
 BASIC SPEED
6.75
 (HT+DX)/4 Basic + Enc.

ENCUMBRANCE
 None (0) = 2xST 16
 Light (1) = 4xST 32
 Med (2) = 6xST 48
 Hvy (3) = 12xST 96
 X-hvy (4) = 20xST 160

PASSIVE DEFENSE
 Armor: 1
 Shield: _____

ACTIVE DEFENSES
 DODGE 6 FARRY 7 BLOCK 5
 = Move KNIFE Weapon/2 Shield/2

DAMAGE RESISTANCE
 Jacket: 1
 T 1

Pl. Cost
ADVANTAGES, DISADVANTAGES, QUIRKS
ABSOLUTE DIRECTION
ACUTE HEARING (+5)
Heart on IT
DOUBLE-JOINTED
DANGER SENSE
ENEMY: Thieves' Guild
roll of 6 or less
OVERCONFIDENCE
POOR

Pl. Cost
SENSITIVE ABOUT HEIGHT
AFRAID OF DROWNING/under water
LOVES HIGH PLACES
NO DRUGS OR ALCOHOL
FLAMBOYANT SHOWOFF
CANNOT READ



REACTION +/- NONE

WEAPONS AND POSSESSIONS

Item	Type	Am.	Level	\$	Wt.
SMALL KNIFE	cut	1d-5	17	30	1/2
	imp	1d-4			
DAGGER	imp	1d-4	17	20	1/2
LOCKPICKS				30	-
CLOTHES (low class)				10	1
SHOES				40	2
LEATHER JACKET				50	4
RING				20	-
SILVER \$				10	-

Totals: \$ 200 Lbs. 7 3/4

WEAPON RANGES

Weapon	SS	ACC	1/2 DMG	MAX
KNIFE	11	0	3	8
DAGGER	12	0	3	8

SKILLS

Skill	Pl. Cost	Level
APER KNOWLEDGE	2	19
CLIMBING	-	19
FAST-TALK	2	12
KNIFE	4	17
KNIFE THROWING	1/2	14
LOCKPICKING	4	13
PICKPOCKET	4	15
SHORTSWORD	4	16
STEALTH	2	15
STREETWISE	2	12
TRAPS (From lockpicking)	1/2	11

SUMMARY

Category	Points	Total
Attributes	85	
Advantages	95	
Disadvantages	-40	
Quirks	-5	
Skills	25	
TOTAL		100

Copyright © 1988 Steve Jackson Games Inc. All rights reserved.

Kuva 1: Alkuperäinen hahmolomake

1.3 Rajaukset

Sovellus tulee pienimuotoiseen yksityiskäyttöön (yksi peliporukka) ja rajoittuu pelkstään GURPSin kolmannen laitoksen perussääntökirjassa mainittuihin sääntöihin ja ominaisuuksiin.

2 Käyttäjäryhmä ja käyttötapaukset

Sovelluksen käyttäjäryhmät ja niiden pääkäyttötapaukset ovat:

Kuka tahansa

- Hahmolomakkeiden tarkastelu

Peruskäyttäjä

- Kirjautuminen
- Salasanan vaihtaminen
- Uuden hahmon luominen
- Hahmon päivittäminen
- Uuden kampanjan lisääminen
- Kampanjan yhteenvetosivun tarkastelu
- Kampanjan päivittäminen
- Kampanjan poistaminen
- Uuden taidon tai ominaisuuden lisääminen
- Vanhan taidon tai ominaisuuden muokkaaminen

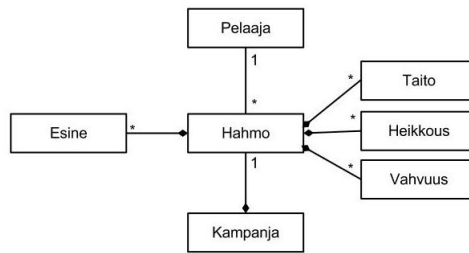
Ylläpitäjä

- Uuden käyttäjän lisääminen
- Käyttäjän poistaminen

Kukin käyttäjäryhmä sisältää myös kaikkien hierarkiassa alempien käyttöryhmien käyttötapaukset, eli peruskäyttäjä pystyy tarkastelemaan hahmolomakkeita siinä missä kirjautumattomatkin käyttäjät. Ylläpitäjällä on omien käyttötapausten lisäksi oikeus tehdä myös kaikki peruskäyttäjän toiminnot.

3 Järjestelmän tietosisältö

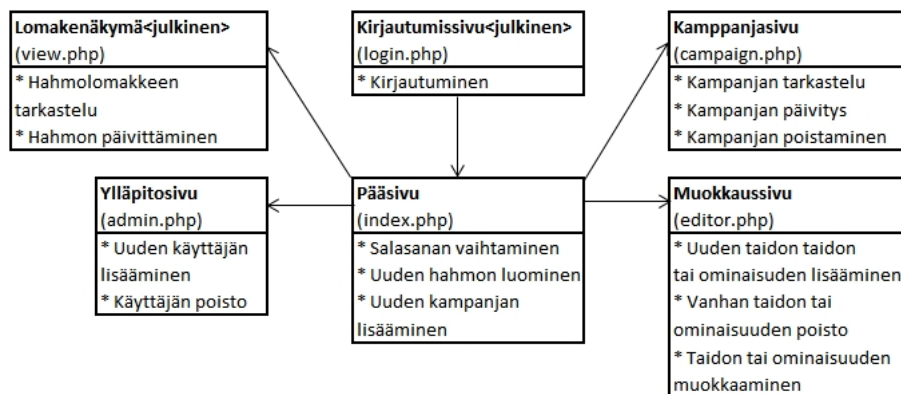
Peruskäsitteiltään kyseessä on varsin yksinkertainen järjestelmä (kuva 2). Oleellisin konsepti on *hahmo*, jolla on tietyt perusominaisuudet (nimi, kuvaus jne.), joiden lisäksi kuhunkin hahmoon liittyy *heikkouksia, vahvuuksia ja taitoja*. Jokaisella hahmolla on myös hallussaan erilaisia *esineitä* joista pelaajan tulee pitää kirjaa. Jokaista hahmoa hallinnoi *pelaaja* eli yksi käyttäjä ja jokainen hahmo voi kuulua yhteenä *kampanjaan*.



Kuva 2: Sovelluksen UML-luokkakaavio

4 Käyttöliittymän hahmotelma

Alustava suunnitelma sovelluksen käyttöliittymälle käyttötapauksineen löytyy kuvasta 3.

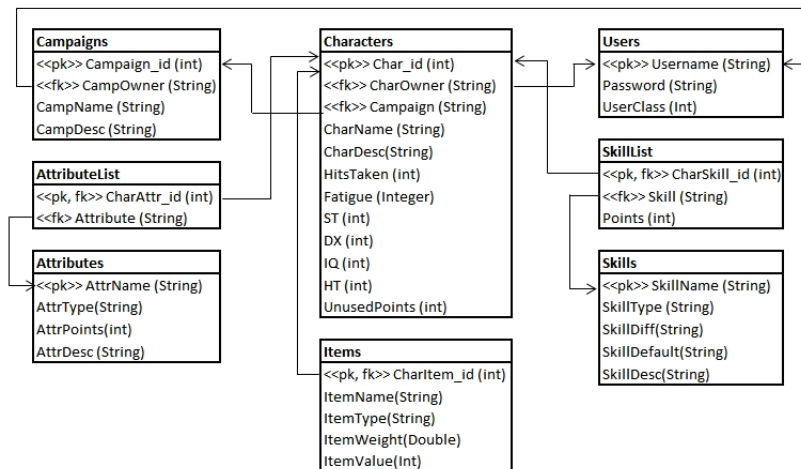


Kuva 3: Sovelluksen käyttöliittymän hahmotelma

5 Relaatiotietokannan rakenne

5.1 Tietokantakaavio

Tietokantataulujen sisältö ja keskinäiset suhteet ovat nähtävissä kuvassa 4.



Kuva 4: Tietokantataulut («pk» pääavain, «fk» viiteavain)

5.2 Tietokannan luontilausekkeet

Tässä esitetään tietokannan taulujen luontilausekkeet MySQL -syntaksin mukaisesti.

Päätaulukot

```

CREATE TABLE Users (
    Username      VARCHAR(20) NOT NULL,
    Password      VARCHAR(50) NOT NULL,
    Userclass     INTEGER     NOT NULL,
    PRIMARY KEY (Username)
);

CREATE TABLE Characters (
    Char_id       INTEGER     NOT NULL AUTO_INCREMENT,
    CharOwner     VARCHAR(20) NOT NULL,
    Campaign      INTEGER,
    CharName      VARCHAR(50) NOT NULL,
    CharDesc      VARCHAR(300),
    HitsTaken     INTEGER,
    Fatigue       INTEGER,
    ST            INTEGER     NOT NULL,
    DX            INTEGER     NOT NULL,
    IQ            INTEGER     NOT NULL,
    HT            INTEGER     NOT NULL,
    UnusedPoints  INTEGER,
    PRIMARY KEY (Char_id),
    FOREIGN KEY (Owner) references Users(Username),
  
```

```

        FOREIGN KEY (Campaign) references Campaigns(Campaign_id)
    );

CREATE TABLE Campaigns (
    Campaign_id    INTEGER    NOT NULL AUTO_INCREMENT,
    CampOwner      VARCHAR(20) NOT NULL,
    CampName       VARCHAR(50) NOT NULL,
    CampDesc       VARCHAR(250),
    PRIMARY KEY (Campaign_id),
    FOREIGN KEY (CampOwner) references Users(Username)
);

CREATE TABLE Skills (
    SkillName      VARCHAR(50) NOT NULL,
    SkillType      VARCHAR(1)  NOT NULL, // (M)ental tai (P) physical
    SkillDiff      VARCHAR(10) NOT NULL,
    SkillDefault   VARCHAR(15) NOT NULL,
    SkillDesc      VARCHAR(500),
    PRIMARY KEY (SkillName)
);

CREATE TABLE Attributes (
    AttrName      VARCHAR(50) NOT NULL,
    AttrType      VARCHAR(1)  NOT NULL, // (A)dvantage tai (D)isadvantage
    AttrPoints    INTEGER    NOT NULL,
    AttrDesc      VARCHAR(500),
    PRIMARY KEY (AttrName)
);

CREATE TABLE Items (
    CharItem_id   INTEGER    NOT NULL,
    ItemName      VARCHAR(50) NOT NULL,
    ItemType      VARCHAR(20),
    ItemWeight    INTEGER,
    ItemValue     INTEGER,
    PRIMARY KEY (CharItem_id),
    FOREIGN KEY (CharItem_id) references Characters(Char_id)
);

```

Yhdistetaulut

```

CREATE TABLE SkillList (
    CharSkill_id  INTEGER    NOT NULL,
    Skills_name   VARCHAR(50) NOT NULL,
    SkillPoints   INTEGER    NOT NULL,
    PRIMARY KEY (CharSkill_id),
    FOREIGN KEY (CharSkill_id) references Characters(Char_id),
    FOREIGN KEY (Skills_name) references Skills(SkillName)
);

```

```
);  
  
CREATE TABLE SkillList (  
    CharAttr_id    INTEGER    NOT NULL,  
    Attr_name      VARCHAR(50) NOT NULL,  
    PRIMARY KEY (CharAttr_id),  
    FOREIGN KEY (CharAttr_id) references Characters(Char_id),  
    FOREIGN KEY (Attr_name) references Attributes(AttrName)  
);
```