

# Seminarska naloga PB za leto 2013

## Travian

Travian je spletna igra, ki jo je razvil Travian Games GmbH. Leta 2006 je bil Travian najboljša spletna igra (igralo jo je več kot 10.000 igralcev) v kategoriji nemške najboljše internetne igre. Je taktična vojaška igra, ki se dogaja v realnem času. Skupaj z dvema angleškima verzijama in z originalno nemško, je bil preveden v več kot 30 jezikov. Danes ima čez 3 milijone igralcev po celem svetu. Narejena je v programskem jeziku PHP in zanjo potrebujemo internetni brskalnik. Bil je prva igra svoje vrste, ki jo je mogoče igrati tudi preko prenosnega telefona. V Sloveniji jo trenutno igra okrog 10.000 igralcev na osmih strežnikih (maksimalno tudi preko 50.000).

## Igralci

Igralec lahko izbira med tremi plemeni:

- Rimljani
- Galci
- Tevtoni

Vsako pleme pa ima tudi svoje posebnosti, prednosti in slabosti. Poleg teh plemen v igri nastopa še sovražno pleme Natarjev in bolj ali manj neprijazna narava.

## Igralno polje

Igralno polje obsega (x,y) koordinate od (-400, -400) do (400, 400). Na vsaki izmed možnih koordinat lahko obstaja naselje, ki pripada določenemu igralcu.

## Alianse

Igra je namenjena skupinskemu igranju – igralci lahko trgujejo, mesta se da okrepiti z vojaki drugega igralca. V igri lahko ustanovimo alianso. Namen le te je, da se lahko več igralcev med sabo združi proti skupnem nasprotniku. V aliansi lahko igralec vidi kolikokrat so napadeni drugi člani alianse. Obstajajo pa tudi posebna komunikativna orodja. V vsaki aliansi je lahko največ 60 članov. Če pa je aliansa večja lahko ustanovi krila, ki se lahko skupaj bojujejo proti drugim (posamezniku ali aliansami). Med sabo lahko pa seveda sklenejo tudi zavezništva, pakte in tudi vojne.

## Travian igralni (spletni) strežniki

Igra se odvija v navideznem svetu, ki ga upravlja igralni strežnik, do katerega igralci dostopajo preko spletnega brskalnika. Za potrebe izdelave zemljevidov in statistik, na vsakem strežniku vsakodnevno objavijo trenutno stanje sveta, ki ga opisuje SQL datoteka z imenom map.sql. Ta datoteka s slovenskega hitrega strežnika (<http://ts1.travian.si/map.sql>) predstavlja osnovo za seminarsko nalogo.

## Opis datoteke map2013.sql

Datoteka vsebuje vsebino tabele x\_world. Ker je datoteka v originalu namenjena MySQL, jo je za bolj splošno uporabo potrebno malenkostno popraviti. V tabeli se nahajajo podatki o posameznih naseljih.

Pomen atributov tabele x\_world je naslednji:

1. **id** - Šifra polja
2. **x** - X koordinata (-400, 400)
3. **y** - Y koordinata (-400, 400)
4. **tid** - Šifra plemena: 1 = Rimljani, 2 = Tevtoni, 3 = Galci, 4 = Narava, 5 = Natarji
5. **vid** - Šifra naselja
6. **village** - Ime naselja
7. **pid** - Šifra igralca
8. **player** - Ime igralca
9. **aid** - Šifra alianse (0 pomeni da igralec ni v aliansi)
10. **alliance** - Ime alianse
11. **population** - Število prebivalcev naselja.

## Uporaba tabele x\_world (pb.fri.uni-lj.si)

Tabela je pripravljena na strežniku. Vsi študenti jo lahko vidite in uporabljate v shemi vaje:

```
use vaje;
describe x_world;
select count(*) from x_world;
ali
describe vaje.x_world;
select count(*) from vaje.x_world;
```

## Kreiranje tabele x\_world (samo za lastno MySQL inštalacijo)

Inštalirajte si najnovejšega odjemalca MySQL Workbench (na učilnici). Zaženite ga, se prijavite v vašo lokalno bazo (npr. vaje), odprite SQL skripte z učilnice in jih poženite:

1. Kreiranje in polnjenje tabele:

File->Open SQL Script

Izberite datoteko map2013.sql (dobite jo na učilnici)

Query->Execute(All or Selection)

2. Počakajte nekaj minut ...

Če je šlo vse v redu, lahko sedaj tabelo `x_world` normalno uporabljate.

## Reševanje nalog in poročilo

Vse opravljeno delo dokumentirajte v obliki poročila, ki vsebuje tekst naloge, rešitev (SQL ali Python koda), prikaz rezultatov in vaše morebitne komentarje. Poročilo oddajte primerno oblikovano v PDF obliki. V tekstu so z zvezdicami (\*) označene nekoliko zahtevnejše naloge.

# Naloge

## 1. Naloga (DDL)

(10%)

Iz tabele `x_world`, opisane z relacijsko shemo

```
x_world(id, x, y, tid, vid, village, pid, player,  
        aid, alliance, population)
```

naredite (CREATE TABLE) in napolnite tabele z naslednjimi relacijskimi shemami in pomeni:

```
pleme(tid, tribe)-- šifra in ime plemena
```

```
aliansa(aid, alliance)-- šifra in ime alianse
```

```
igralec(pid, player, #tid, #aid)-- šifra in ime igralca, njegovo pleme in  
njegova aliansa
```

```
naselje(id, x, y, vid, village, population, #pid)-- šifra polja, x in y  
koordinati, šifra vasi, ime vasi, populacija, šifra igralca lastnika vasi
```

**Pri reševanju naslednjih nalog uporabljajte SAMO tabele iz prve naloge, pa še to v minimalnem potrebnem obsegu. Uporaba originalne tabele `x_world` ni dovoljena!**

## 2. Naloga (DDL)

(30%)

- Naredite pogled `x_view`, ki bo iz novih tabel naredil pogled ekvivalenten tabeli `x_world` (CREATE VIEW).
- S pomočjo SQL poizvedb preverite, ali sta vsebini `x_view` in `x_world` identični.
- \*S pomočjo ustreznih DDL ukazov ustvarite tabelo `top10(alliance, SteviloNaselij)`, ki hrani alianse z največ naselji. Dodajte bazne prožilce tako, da se bo lista posodabljala vsakič ko nekdo zgradi novo naselje.

### 3. Naloga (DML)

(30%)

Nad dobljenimi tabelami iz naloge 1 v jeziku SQL napišite poizvedbe, s pomočjo katerih boste lahko odgovorili na naslednja vprašanja.

- a) Kateri igralec ima največje naselje?
- b) Koliko igralcev ima nadpovprečno veliko naselje?
- c) Izpišite podatke o vseh naseljih igralcev brez alianse, urejeno padajoče po x in nato y koordinati.
- d) Katero pleme je najštevilčnejše (glede na populacijo)?
- e) Izpišite število nadpovprečno močnih alians (povprečje populacije računajte glede na alianse, ne na vse igralce).
- f) \*Igralec BAM-BAM želi preimenovati vsa svoja naselja na naslednji način. Uredil jih bo po populaciji, najmočnejše bo »X-00«, naslednje »X-01« in tako dalje. Nalogo lahko rešite v več korakih (zaporedju poizvedb).
- g) Napišite shranjen podprogram, ki za poljubne koordinate (parametra x in y) vrne populacijo na največ podani razdalji (parameter razdalja). Npr. razdalja 10 pomeni vse koordinate od vključno (x-10, y-10) do (x+10, y+10). Za preverjanje robnih pogojev (koordinate izven [-400,400] po potrebi uporabite IF stavek.
- h) Izpišite imena igralcev, ki imajo vsa svoja naselja na območju x, ki je med 100 in 200 in y, ki je med 0 in 100.
- i) Pospešite izvajanje počasnih poizvedb s kreiranjem ustreznih indeksov (CREATE INDEX). Svoje predloge dokumentirajte s faktorji pohitritve in jih utemeljite.
- j) Poiščite igralce, ki imajo umirajoče naselje. Za umirajoče naselje vzemite tista naselja, ki imajo manj kot 3% povprečne populacije igralca (povprečna populacija igralca je populacija igralca ulomljeno s številom njegovih naselij) .

### 4. Naloga (ODBC)

(20%)

V programskem jeziku Python napišite program, ki se priključi na podatkovno bazo in za celotno igralno polje izračuna gostoto populacije in gostoto populacije določene alianse. Gostoto računajte na območjih velikosti 10x10 polj po formulah:

$$Gostota\ populacije = \frac{Skupna\ populacija\ na\ območju}{100}$$

$$Gostota\ alianse = \frac{Skupna\ populacija\ alianse\ na\ območju}{100}$$

Rezultate izračunane gostote (za vsako izmed 80x80=6400 območij) shranite v primerne tabeli gostotaPopulacije in gostotaAlianse. Za alianso izberite najmočnejšo alianso glede na populacijo.

## 5. Naloga (ODBC)

(10%)

Če znate, rešujte (a), sicer (b). (a) vam prinese dodaten bonus. Grafi so lahko dvodimenzionalni (višja intenziteta točke pomeni večje število) ali tridimenzionalni (stolpčni). Vsekakor naj bodo dovolj nazorni.

- a) **\*(Bonus dodatnih 10% za demonstracijo)** V Pythonu napišite GUI aplikacijo (Qt ali podobno), ki se priključi na podatkovno bazo in v obliki grafov izriše rezultate izračunane gostote poselitev iz četrte naloge. V okviru te naloge lahko realizirate tudi celotno četrto nalogo, brez shranjevanja vmesnih rezultatov.
- b) Iz programa Microsoft Excel se priključite na podatkovno bazo in v obliki grafov izrišite rezultate izračunane gostote poselitev iz četrte naloge.