17. W e g a s ( Web Game System ) -S

Vicol Robert- Ilie

Mazilu George Viorel

Moroșanu George Cosmin

Minuți Vlăduț- Ștefan

Facultatea de Informatică din Iași, Alexandru Ioan Cuza, Iași

[robert.vicol@fenrir.info.uaic.ro](mailto:mihai.minzatanu@fenrir.info.uaic.ro)

[mazilu.george07@gmail.com](mailto:mazilu.george07@gmail.com)

[minutivlad@gmail.com](mailto:minutivlad@gmail.com)

[georgecosminmorosanu@gmail.com](mailto:georgecosminmorosanu@gmail.com)

**Abstract.** În acest document oferim detalii cu privire la aplicația noastră Bomberman. Proiectul include atât tehnologii front-end cât și back-end, utilizatorii putând accesa o interfață care le permite logarea, delogarea, verificarea contului, construirea robotului și participarea la jocul propriu-zis.

**Cuvinte-cheie:** Aplicație client-server / Bază de date / Tehnologii Web / Transfer de date / Concurență /

1. Descrierea aplicației

Proiectul se constituie dintr-o aplicație web, un joc la care pot participa mai multi utilizatori. Jocul se numește Bomberbot. Include micro-management de resurse, de diferite categorii, ce urmează a fi detaliate în bazele de date expuse. Scenariul constă în alegerea unui teren de bătălie si a unor resurse pentru construirea unui robot de luptă. Fiecare player va avea o sumă de bani alocată, pe care o poate folosi pentru a cumpăra resurse cu care pot imbunătăți robotul. Conflictul scenariului este între playeri, aceștia putând caștiga doar în urma distrugerii robotului inamicului / inamicilor. Un joc este considerat egal dacă toți roboții activi sunt distruși in același timp. Clienții trebuie sa fie logați pentru a accesa funcționalitățile aplicației.

1. Tehnologii utilizate
   1. Comunicarea

În etapa curentă considerăm util protocolul HTTP pentru conectarea clienților la server. De asemenea, este important ca pachetele de date sa nu se piardă, să ajungă în întregime la destinatar Protocolul HTTP satisface toate aceste cerințe la nivelul unei aplicații web.

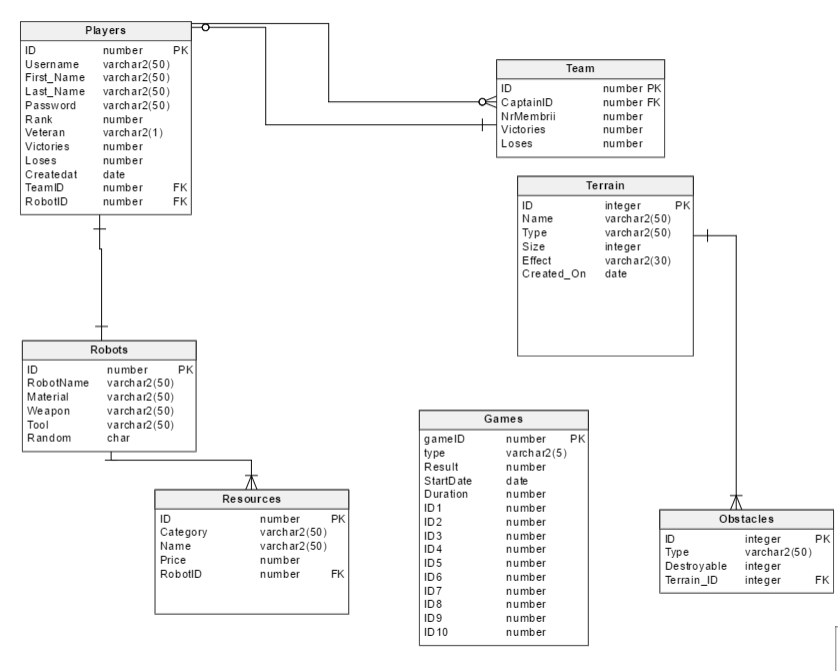
* 1. Limbaj folosit

Vom folosi pentru realizarea aplicației limbajele C, JAVA, SQL, DDL, DML, DCL.

* 1. Unity / Unreal Engine

Este posibil ca în realizarea jocului, să ne putem folosi de aceste programe.

* 1. Diagrama a bazei de date



1. Baze de date

Avem in baza de date formată 7 tabele. Fiecare stochează informații importante pentru aplicația noastră:

1. Tabela Players conține informațiile completate de utilizator în interfață, la momentul creării contului, precum și date despre meciurile jucate, robotul curent (ID-ul său, vechimea jucătorului (dacă este veteran sau nu – are 100 de meciuri jucate sau nu). Jucătorul poate alege să se înscrie într-o echipă, moment în care va fi completat câmpul TeamID. Aplicația va avea un sistem de ranking (ladder system). Fiecare jucător va porni cu 0 puncte. La fiecare victorie va primi 25 de puncte, iar la fiecare înfrângere va pierde 25 de puncte. Rankul nu poate sa scadă, în schimb, sub 0.
2. Tabela Resources validează micromanagementul resurselor, evidențiază prezența unor modalități diferite de a construi un robot în funcție de materialele de construcție, de uneltele și de armele disponibile. Câmpul categorie imparte resursele în arme, unelte, și materiale de construcție. Fiecare resursă are un cost, care este plătit la începutul jocului de player. Modul în care fiecare își consumă banii disponibili constituie strategia necesară pentru victorie. Exemple de materiale: lemn, titanium, otel Exemple de arme: Plebeus\_bombs, Proximity\_mines, Poison\_bombs Exemple de unelte: Jetpack, Detonator, Scutantiexplozie, etc
3. Tabela Robots: Fiecare robot are propriul său ID și nume. De asemenea, fiecare robot trebuie sa aibă în compoziția sa un material de construcție, o armă și o unealtă. Uneltele au proprietăți diverse, reprezintă bonusuri sau abilități în general defensive. Câmpul random poate avea 2 valori, y/n. Jucătorii care nu au achievementul VETERAN, nu pot juca decât cu roboți construiți în mod random (nu-și pot modela robotul).
4. Tabela Terrain conține terenurile disponibile pentru joc, fiecare având un ID, un nume, precum și un tip (junglă, zonă urbană, vulcan, pol nord, etc). Mărimea hărții este dată de Size, harta fiind o matrice de dimensiunea Size x Size. De asemenea, tipul hărții poate avea un efect asupra jocului, spre exemplu gheața îl poate face pe robot sa alunece, lava îl poate răni, etc.
5. În tabela Obstacles se găsesc toate obstacolele care se află pe teren, acestea având câte un ID unic și un tip (copac, cărămidă, lemn, bloc de gheață, etc), precum și ID-ul hărții de care aparțin. Dacă obstacolul poate fi distrus de o bombă, atunci câmpul Destroyable va avea valoarea ‘y’, altfel ‘n’.
6. Tabela Team conține un ID al echipei, ID-ul căpitanului, numărul de membri, precum și numărul de victorii, respectiv de înfrângeri.
7. Tabela Games conține un istoric al jocurilor, reține câștigătorul, durata, tipul, cât și participanții, în funcție de care se vor distribui punctaje pentru rank.

Există constrângeri în tabele după cum urmează:

* Câmpul Rank, așa cum am explicat, nu poate avea valori negative. În schimb, acesta va fi un multiplu de 25.
* În câmpul Category din tabela Resources găsim unul din aceste 3 stringuri: Materials, Weapons, Tools.
* Din motive evidente, câmpul Price din tabela Resources nu poate avea valori negative.
* Veteran si Random sunt câmpuri ce stochează valori asemănătoare unui tip bool.
* Numărul de membri ai unei echipe nu poate fi mai mare decât 5.
* Durata unui joc nu poate depăși 3 minute.
* Result conține fie ID-ul câștigătorului, fie 0 (în cazul în care meciul s-a terminat la egalitate)

4 Bibliografie

https://en.wikipedia.org/wiki/Internet\_protocol\_suite

https://www.w3schools.com/sql/sql\_create\_db.asp