WordGuessr

Documentatie

Obiectivul temei

O aplicație de gamification, care să permită utilizatorilor

să câștige “tokens”, “badges” și să existe un sistem de “ranking” în baza rezolvării unor

quest-uri. Quest-urile pot fi propuse de oricine, cu condiția să aibă suficienți

“tokens”/”badges”/”ranking” pentru a recompensa eventualii câștigători/participanți.

Modul principal de a câștiga “tokens” este prin rezolvarea quest-urilor.

Solutia aleasa

O aplicatie pentru un joc de ghicit cuvinte bazat pe o descriere a cuvantului , in care cuvintele sunt propuse de orice utilizator iar raspunsurile corecte sunt contabilizate pentru a realiza un clasament intre toti jucatorii existenti.

Limbajul si framework ul ales

Pentru modelarea acestei cerinte am folosit limbajul orientat obiect Java si framework ul java swing pentru realizarea interfetei grafice pentru utilizatori , iar pentru persistenta datelor am folosit MySql.

Proiectare

Pentru a realiza proiectarea aplicației folosind modelul MVP (Model View Presenter):

* **model** – cuprinde clasele care definesc datele care trebuie afișate sau asupra cărora se acționează in alt mod in interfața grafica .
* **view** – cuprinde clasele care afișează datele (modelele) si direcționează comenzile/evenimentele către controller pentru a acționa asupra datelor
* **controller**– cuprinde clasele care acționează asupra modelelor , dar si asupra interfețelor grafice din sub pachetul **view**

Implementare

Pachetul Model – dupa cum am precizat anterior pachetul Model cuprinde clasele care definesc datele care trebuie afișate sau asupra cărora se acționează in alt mod in interfața grafica aceste clase sunt:

* User : clasa utilizata pentru a stoca atributele unui utilizator necesare atat pentru logare cat si pentru ai calcula punctajul total, clasa user are atributele:
  + id,
  + username,
  + words(numarul de cuvinte ghicite),
  + firsttryBadges(numarul de badges de tipul first try),
  + inarowBadges(numarul de badges de tipul in a row),
  + found(precizeaza daca cuvantul ales de utilizator a fost gasit de ceilalti utilizatori),
  + desc(descrierea cuvantului ales).

Ca metode clasa User are doar gettere si settere.

* Attemp : clasa utilizata pentru a stoca datele referitoare la cine a incercat fiecare provocare (statistica necesara pentru acordarea de badgeuri).Aceasta clasa are doar 3 atribute: id,idChallenger,idChallenge.
* ConnectionFactory : clasa responsabila pentru crearea si inchiderea de conexiuni la baza de date , aceasta are doar 2 metode CreateConnection si CloseConnection.
* DbAttemptPersistance: aceasta clasa se ocupa cu persistenta datelor despre incercari din baza de date, este responsabila pentru a genera diferite query uri care vor fi aplicate pe baza de date pentru a obtine si modifica date din baza.
* DbUserPersitance: similar cu clasa de mai sus , aceasta se ocupa cu persistenta userilor si este putin mai complexa , aceasta clasa se ocupa cu generarea si aplicarea de query uri pentru baza de date, aceasta are atat metode care obtin informatii din baza de date pentru a fi afisate pe interfata grafica dar si care fac modificari pe baza de date pentru a realiza functionalitatea jocului.

Pachetul View

Pachetul view contine 2 clase : LoginView si MainView. Aceste clase reprezinta interfata grafica pentru utilizator , acestea sunt generate cu un window builder asadar numarul de atribute ale acestor clase este unul mare( butoane , labele, listenere etc). De asemenea aceste clase contin niste metode pentru a actualiza informatiile afisate , acestea vor fi apelate de controllere la momentul apasarii butoanelor.

Pachetul Controller

La fel ca pachetul view acest pachet contine 2 clase LoginController si MainController acestea sunt controllerele pentru cele 2 tipuri de viewuri pe care le are aplicatia.

Descrierea jocului

Joc de ghicit cuvinte bazat pe o descriere a cuvantului , in care cuvintele sunt propuse de orice utilizator iar raspunsurile corecte sunt contabilizate pentru a realiza un clasament intre toti jucatorii existenti.

Functionalitati:

-orice utilizator poate sa isi vada statisticile dar si clasamentul general

-orice utilizator poate sa incerce sa ghiceasca cuvintele propuse de ceilalti utilizatori(ghicirea unui cuvant este rasplatita cu +1 cuvant, iar daca cuvantul este ghicit din prima incercarea se adauga un firstTryBadge), odata ghicit cuvantul provocarea devine inactiva(doar o persoana poate rezolva fiecare quest).

-orice utilizator care are cel putin 1 cuvant ghicit poate sa propuna un cuvant( costul pentru propunerea unui cuvant este -1) , gresirea raspunsului de catre 10+ utilizatori ii va aduce utilizatorului care a propus inarowBadges in felul urmator ( la a 10 lea greseala va primi un Badge iar apoi pentru fiecare utilizator urmator care greseste inca un badge), odata ghicit cuvantul se poate propune altul.

- totalul unui utilizator este realizat folosind formula : cuvinteGhicite + 5\*firstTryBadges + 3\*inaRowBadges.

Posibile dezvoltari

Posibilele dezvoltari ale acestui joc ar fi implementarea unui register pentru utilizatori (momentan doar utilizatorii introdusi in baza de date pot sa participe), adaugarea mai multor tipuri de badge-uri , adaugarea unei constrangeri de timp (de exemplu un utilizator poate sa propuna doar un cuvant pe zi) sau a unor constrangeri legat de numarul de incercari zilnice ale unui utilizator, sau implementarea unui „word of the day” pe care l-ar putea ghici fiecare utilizator.