

JsGames

Зорница Атанасова Костадинова, 4 курс, КН, фн: 80227,
Искрен Ивов Чернев, 4 курс, КН, фн: 80246

29 януари 2011 г.

Цел на проекта

Настоящият документ е курсова работа към проекта “JsGames” по предмета “WWW Технологии”. Описали сме задачата която си поставихме с този проект, решението, базата данни (схема, описание на таблиците и релациите между тях), интерфейса и възможностите за разширяване на проекта. Обяснили сме методите и основните архитектурни принципи залегнали в разработката на проекта, както и използваните технологии.

Съдържание

1	Описание на проекта	3
2	Използвани технологии	3
2.1	Haml	3
2.2	Sass	3
2.3	Ruby on Rails Framework	4
2.4	NodeJs	4
2.5	jQuery	4
2.6	Документация	4
2.7	Yui3 library	4
2.8	Mercurial	4
2.9	Zero CSS	4
2.10	Ruby Gems	4
2.11	Статистика за проекта	4
2.12	Цел	4
3	Реализация	4
3.1	Програмен език	4
3.2	Дизайн	5
4	Инсталация	6
4.1	Инсталиране на зависимости	6
4.2	Заклучение	6

1 Описание на проекта

Проект: сайт с javascript игри.¹

- сайта ще поддържа потребители със следната информация за тях: – email; – парола; – nickname.
- информация за игра: – име; – кратко описание; – пълно обяснение на правилата.
- за всяка изиграна игра ще се пази следната информация: – коя е играта; – потребители играли играта (може някой да са компютри, в такъв случай се пази трудността на компютъра); – резултат; – продължителност на играта; – ако играта не бъде изиграна до 1 ден ще се изтрива информацията за нея.
- възможност за изкарване на класиране: – най-много изиграни игри - общо за всички игри или за конкретна игра; – най-много изкарани точки - общо точки или средно аритметично, общо и за конкретна игра; – най-дълго прекарано време в игри - общо време или средно аритметично, общо или за конкретна игра.
- ще има възможност за динамично добавяне, редактиране, изтриване на горепосочените данни където това има смисъл
- допълнителни пояснения за игрите: – игрите ще бъдат имплементирани на javascript; – всички игри ще имат изкуствен интелект (имплементиран като javascript клиент) с поне една степен на трудност; – сървърът ще поддържа комуникация между различни javascript клиенти (при игра на няколко души) за да се обменят изиграните ходове; – ще може да се играе и само от един клиент (браузър) ако единия играч е човек, а другия компютър (изкуствен интелект).

2 Използвани технологии

2.1 Haml

Haml е [1] Haml is a markup language that's used to cleanly and simply describe the XHTML of any web document, without the use of inline code. Haml functions as a replacement for inline page templating systems such as PHP, ERB, and ASP. However, Haml avoids the need for explicitly coding XHTML into the template, because it is actually an abstract description of the XHTML, with some code to generate dynamic content.

2.2 Sass

Sass [2] -> Css [3] Sass is an extension of CSS3, adding nested rules, variables, mixins, selector inheritance, and more. It's translated to well-formatted, standard CSS using the command line tool or a web-framework plugin.

Sass has two syntaxes. The new main syntax (as of Sass 3) is known as "SCSS" (for "Sassy CSS"), and is a superset of CSS3's syntax. This means that every valid CSS3 stylesheet is valid SCSS as well. SCSS files use the extension .scss.

¹Проектът се хоства на <http://iskren.info:50005/>.

The second, older syntax is known as the indented syntax (or just “Sass”). Inspired by Haml’s terseness, it’s intended for people who prefer conciseness over similarity to CSS. Instead of brackets and semicolons, it uses the indentation of lines to specify blocks. Although no longer the primary syntax, the indented syntax will continue to be supported. Files in the indented syntax use the extension .sass.

2.3 Ruby on Rails Framework

2.4 NodeJs

2.5 jQuery

2.6 Документация

2.7 Yui3 library

2.8 Mercurial

2.9 Zero CSS

2.10 Ruby Gems

2.10.1 База от данни

PostgreSQL [9]

2.11 Статистика за проекта

- Използвани езици за програмиране
- Редове код от началото на проекта
- Брой тестове
- Брой commits в системата за контрол на версиите

2.12 Цел

3 Реализация

3.1 Програмен език

Ruby [5] and Javascript [6]. Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.

3.1.1 Уеб сървъри

За работата на продукта е нужен web server [4].

Използваме:

- **Webrick** [7] - web server който се ... WEBrick is a Ruby library providing simple HTTP web server services. The server also provides code for simple server services other than HTTP. It is used by the Ruby on Rails framework to test applications in a development environment.
- **NodeJs** [8] - web server ...

3.1.2 База от данни

За коректна работа, Drupal се нуждае от релационна база от данни. Връзката с БД става чрез модул който имплементира определено API. Това позволява независимост на избора на БД от останалите компоненти на продукта. В момента се поддържат две от най-известните бази с отворен код - PostgreSQL [9].

3.2 Дизайн

Спазени са основните принципи на Обектно Ориентираното Програмиране [10] - капсулиране, полиморфизъм, наследяване, абстракция, обекти.

3.2.1 Шаблони за дизайн

Следните шаблони за дизайн (design patterns) са използвани в JsGames:

- **singleton** - Ако разглеждаме модулите и темите като обекти, то всички те реализират the singleton pattern.
- **model-view-controller** - ...
- **event-driven approach** - ...

NodeJs Node's goal is to provide an easy way to build scalable network programs. In the "hello world" web server example above, many client connections can be handled concurrently. Node tells the operating system (through epoll, kqueue, /dev/poll, or select) that it should be notified when a new connection is made, and then it goes to sleep. If someone new connects, then it executes the callback. Each connection is only a small heap allocation.

This is in contrast to today's more common concurrency model where OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use.

User

View

4 Инсталация

4.1 Инсталиране на зависимости

За да разработим JsGamesци инсталирахме:

1. ruby
2. rails
3. node
4. socket.io
5. npm
6. PostgreSQL
7. yui3
8. jsLint
9. Mercurial
10. more ... :)

4.2 Заключение

Література

- [1] <http://haml-lang.com/>
- [2] <http://sass-lang.com/>
- [3] <http://en.wikipedia.org/wiki/Css>
- [4] http://en.wikipedia.org/wiki/Web_server
- [5] <http://www.ruby-lang.org/en/>
- [6] <http://en.wikipedia.org/wiki/ECMAScript>
- [7] <http://en.wikipedia.org/wiki/WEBrick>
- [8] <http://nodejs.org/>
- [9] <http://www.postgresql.org/>
- [10] http://en.wikipedia.org/wiki/Object-oriented_programming