

# 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

<b>1</b>	<b>Описание на проекта</b>	<b>3</b>
<b>2</b>	<b>Използвани технологии</b>	<b>4</b>
2.1	Haml . . . . .	4
2.2	Sass . . . . .	4
2.3	Ruby on Rails Framework . . . . .	4
2.4	NodeJs . . . . .	5
2.5	jQuery . . . . .	5
2.6	Yui3 library . . . . .	5
2.7	Mercurial . . . . .	5
2.8	RubyGems . . . . .	6
2.9	Статистика за проекта . . . . .	6
2.10	Цел . . . . .	6
<b>3</b>	<b>Реализация</b>	<b>6</b>
3.1	Програмен език . . . . .	6
3.2	Дизайн . . . . .	7
<b>4</b>	<b>Инсталация</b>	<b>8</b>
4.1	Инсталиране на зависимости . . . . .	8
4.2	Заклучение . . . . .	8

## 1 Описание на проекта

Проект: сайт с javascript игри.<sup>1</sup>

- сайта ще поддържа потребители със следната информация за тях:
  - email
  - парола
  - nickname
- информация за игра:
  - име
  - кратко описание
- 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
- 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 (браузъри) при игра на няколко души за да се обменят изиграните ходове
  - ще може да се играе и само от един клиент (браузър) ако единият играч е човек, а другият компютър (изкуствен интелект)

---

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

## 2 Използвани технологии

### 2.1 Haml

Haml е [1] маркър език, с който може чисто и просто да се описва XHTML за всеки онлайн документ, без вграждане на код. Haml е предназначен като алтернатива на други шаблонни езици, които вграждат код, като PHP, ERB и ASP. Също така с Haml избягвате писането на чист XHTML с използването на семантична идентация.

### 2.2 Sass

Sass [2] е добавка към CSS3 [3], в която е добавено влагане на правила, променливи, модули, наследяване на селектори и други. Sass се трансформира до добре форматиран CSS с помощта на конзолна програма или плъгин към уеб фреймуърк.

Sass има два синтаксиса. Новия синтаксис (въведен от Sass 3) още известен като SCSS е надмножество на CSS3. Това означава, че всеки валиден CSS3 файл е също валиден SCSS. SCSS файловете имат разширение `.scss`.

Втория, по-стар синтаксис е известен като идентифицирания синтаксис (или просто "Sass"). Вдъхновен от краткостта на Haml, той е предназначен за хора, които предпочитат изразителността пред близостта с CSS. Вместо точка и запетая и скоби се използва семантична идентация, за да се обособяват блокове. Въпреки че вече не е основния синтаксис, ще продължи да бъде поддържан. Файловете с идентифициран синтаксис имат разширение `.sass`.

### 2.3 Ruby on Rails Framework

Ruby on Rails, често съкращаван като Rails или Ror, е уеб фреймуърк с отворен код за програмния език Ruby. Той е предназначен за програмиране по Agile методологията, която се използва от уеб програмисти за ускорено програмиране.

Като много уеб фреймуърци, Ruby on Rails използва Model-View-Controller (MVC) архитектура за организиране на приложението.

Ruby on Rails включва програмки, които правят често срещани задачи в програмирането по-лесни, като например автоматичното генериране на шаблонен код, за по-бързо стартиране на модел или изглед. Също включен е WEBrick - прост уеб сървър писан на Ruby, както и Rake - система за билдване (като make) която се дистрибутира като gem (пакет за Ruby). Заедно с Ruby on Rails тези програмки предоставят базова среда за програмиране.

Ruby on Rails разчита на уеб сървър за да работи. Mongrel е предпочитан пред WEBrick, но също може да се ползва Lighttpd [5], Abyss [6], Apache [7], nginx [8] и много други. От 2008 уеб сървърът Passenger [9] заменя Mongrel като препоръчителна опция. Ruby on Rails е известен също и с използването на javascript библиотеките Prototype [10] и Script.aculo.us за Ajax [11]. В началото Ruby on Rails използва лек SOAP, но по-късно е заменен от REST. От версия 3 Ruby on Rails използва Unobtrusive javascript [12] – техника за разделяне на логиката от изгледа на уеб страницата.

## 2.4 NodeJs

Целта на Node.js [20] е да предостави лесен начин за програмиране скалируеми, работещи в мрежа програми. Node инструктира операционната система да го уведомява за нови клиенти и после заспива. Ако някой се свърже се изпълнява предварително зададена функция (callback). За всяка връзка се използва съвсем малко памет.

Това е в контраст с по-популярния модел на едновременна работа, в който се използват нишки. Мрежово програмиране имплементирано с нишки е сравнително неефективно и много трудно за използване. Node използва доста по-малко памет при високо натоварване, от колкото системи, които заделят по 2MB за всяка нишка / конекция. Още повече потребителите на Node няма нужда да се притесняват от dead-lock - просто защото няма заключване. Няма функции в Node, която директо изпълнява вход/изход - затова процеса никога не блокира. Поради този факт непрофесионалисти могат да пишат бързи системи.

## 2.5 jQuery

jQuery [13] е javascriptбиблиотека предвидена да улесни скриптирането на HTML от страна на клиента. Тя е пусната през януари 2006 от Джон Ресиг. Използвана е в 41% от 10000 най-посещавани сайтове. Това е най-популярната javascriptбиблиотека в момента.

jQuery е безплатна, с отворен код, лицензирана под MIT и GNU v2 лицензите. Синтаксиса на jQuery е предвиден да улесни навигацията в документа, селектирането на DOM обекти, създаването на анимации, обработката на събития и разработването на Ajax приложения. jQuery също предоставя възможност за писане на плъгини. С използването на тези възможности програмиста може да създаде абстракции за анимация и взаимодействие от ниско ниво, сложни ефекти както и компоненти от високо ниво поддържащи теми. Това подпомага за създаването на сложни и динамични уеб страници.

## 2.6 Yui3 library

YUI [14] е javascriptбиблиотека разработвана от Yahoo!. Тя има добре развита модулна система, която позволява да се сваля само кода от библиотеката, който реално се използва. Системата ѝ за събити надгражда тази дефинирана в брауъра и позволява по-лесна обработка и добавяне на специализирани събития.

## 2.7 Mercurial

Mercurial [15] е дистрибутирана система за управление на сорс код.

Традиционните системи за управление на сорс като Subversion са с типични клиент-сървър архитектури. Те имат централен сървър който пази информацията за различните ревизии на проекта. В контраст Mercurial е напълно дистрибутирана, позволявайки на всеки програмист да има локално копие на цялата история на проекта. Това означава, че програмистите са независими от връзката си с интернет или централния сървър. Добавянето, разклоняването и сливането в проекта са лесни и бързи операции.

## 2.8 RubyGems

RubyGems [16] е пакетен мениджър за програмния език Ruby, който предоставя стандартен формат за разпространението на Ruby програми и библиотеки във самодостатъчен файлов формат `.gem`. Той предоставя конзолен интерфейс за лесна инсталация и управление на гемове, както и сървър за тяхното разпространение. RubyGems е аналогичен на EasyInstall за програмния език Python. В момента RubyGems е част от стандартната библиотека на Ruby версия 1.9.

### 2.8.1 База от данни

PostgreSQL [21] е обектно-релационна система за управление на база от данни. Тя е лицензирана под MIT лиценз и следователно е безплатна и с отворен код. Както други системи с отворен код PostgreSQL не се контролира от нито една единствена компания – интернационална група от програмисти и компании разработват продукта.

## 2.9 Статистика за проекта

- Използвани езици за програмиране
  - javascript
  - Ruby
  - SQL
  - html (генериран чрез haml)
  - css (генериран чрез sass)
- Редове код от началото на проекта - 5000
  - Ruby 1200
  - javascript 3200
- Брой тестове
  - Ruby 14
  - javascript 30
- Брой commits в системата за контрол на версиите - 130

### 2.10 Цел

## 3 Реализация

### 3.1 Програмен език

Ruby [17] and Javascript [18]. 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** [19] - 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** [20] - 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 [21].

## 3.2 Дизайн

Спазени са основните принципи на Обектно Ориентираното Програмиране [22] - 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



## Литература

- [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](http://en.wikipedia.org/wiki/Web_server)
- [5] <http://www.lighttpd.net/>
- [6] <http://www.aprelum.com/>
- [7] <http://www.apache.org/>
- [8] <http://nginx.org/>
- [9] <http://www.modrails.com/>
- [10] <http://www.prototypejs.org/>
- [11] [http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- [12] [http://en.wikipedia.org/wiki/Unobtrusive\\_JavaScript](http://en.wikipedia.org/wiki/Unobtrusive_JavaScript)
- [13] <http://jquery.com>
- [14] <http://developer.yahoo.com/yui/>
- [15] <http://mercurial.selenic.com/>
- [16] <http://rubygems.org/>
- [17] <http://www.ruby-lang.org/en/>
- [18] <http://en.wikipedia.org/wiki/ECMAScript>
- [19] <http://en.wikipedia.org/wiki/WEBrick>
- [20] <http://nodejs.org/>
- [21] <http://www.postgresql.org/>
- [22] [http://en.wikipedia.org/wiki/Object-oriented\\_programming](http://en.wikipedia.org/wiki/Object-oriented_programming)