

Team Project: 2ID60 / Web Technology

GAMR



Created by:

- Koen Klaren
- Tim Segers
- Thomas Musters
- Ky-anh Tran

Responsible professor:

Dr. K. Verbert

Instructors:

Dr. N. Stash

Dr. G.H.L. Fletcher

Final report of group project; 70% of final grade

Motivation

At first we had no clue what to make, and we only came up with rip-offs of the most obvious applications, like amazon, last FM or IMDb.

However since we created our group on Facebook in the Software Science & Web Science group, we knew we all liked computers, and after a while Tim came up with the suggestion that we probably all liked games. From there, we asked everyone if they agreed this would be what our application would be about, and we all agreed.

The purpose of our application is simple, provide information about games so players can make a more educated choose, and perhaps even suggest new games based on their current interests. The latter turned out to be rather difficult, since we would need a lot of user data. The application would have needed thousands of recorded visitors before an algorithm to suggest new games could be developed.

For the USE cases, I'm pretty sure we didn't use any. We wanted to think of something that was completely new for all of us, which would not be the case if we used an existing USE case. If anything, we should write a USE case report of our own project for others to use.

Data

As for the data, we chose to use sources some of us were already familiar with, because we figured it would be easier to apply the data in a way that made sense.

For our information we mainly use GiantBomb
API can be found here <http://www.giantbomb.com/api/>

We retrieve a score from MetaCritic, of which no official API is released, but we managed to find a decent unofficial one here, which is still in BETA phase however: <https://www.mashape.com/byroredux/metacritic>

The third source we used was youtube, which we only embedded, but of which the API can be found here: <https://developers.google.com/youtube/>

We intended to use other sources as well, but this revealed multiple problems:

- Data overlap
- Hard to interlace from different sources
- No API available
- Copyright infringement

Technical architecture

We use a simple REST framework called Slim. All HTTP requests that are not for static resources, are passed to this framework, which will then perform some routing logic on the request and return the correct response.

From within our route definitions, we call our API helper methods to retrieve data from our sources. The home- and gameinfo routes respond with an HTML5 page, the search-results route however returns a JSON document, which is used by our Javascript to provide dynamic search results.

User data

For our user tracking functionality we decided that the most useful information for the user would be suggestions that aid the user in discovering new games. We display similar games on the page that displays the information on a game the user searches for. This information is obtained from the Giantbomb database, and thus subject to constant change. This means this information is kept up to date as new similar games release.

To present the user with a more general suggestion as what to discover, we present them with the five latest searches of other users right on the homepage. They can click these suggestions to land on the info page of the title they clicked. We achieve this by saving all searches of our users in an SQLite3 database, and at the time Gamr is loaded, the web app queries the database for the five entries with the highest timestamp, and displays those results on the page.

This gives the user the possibility to gauge what other users are interested in and suggests them titles that are in a broader range of genres than the ones found under similar games.

Reflection

The way our project is related to the topics taught in this course, in the most basic way, the course is called Web Technology, and we created a Web Technology.

Being a little more specific, first of all, we used what we learned from the HTML, CSS, REST and SOAP tutorials, some to more extend then others, to create our application, of course combining this with the other knowledge we already had on these topics from other, previous courses. The biggest overlap between the courses and our project must be the use of web services to get data, like the info from GiantBomb.

But next to what we learned in the courses, we also had to learn some things on our own, which has mainly been done by Tim, picking up PHP in a few weeks, enough to create a sustainable Web Technology. Since Koen already works with Websites, he didn't have to pick up much before he could start on the HTML/CSS of the code. Use of the different API's also had to be understood, but this required less effort since there was more than enough documentation and most examples we found were similar to what we planned to do with the API's. And even though it might not be course-related, one of the most important things we learned was planning to hold meetings on Facebook doesn't work.

Effort

At first, we thought this was going to be be easy and wouldn't require that much time, that is why we only had a vague concept of what we wanted to make when the first presentation came by. This showed us two things. First, that there was a lot of effort in other groups projects, and second, that most groups were ahead of us. This caused us to shift into a higher gear to catch up, but we ran into the same problem most student do when they found out a course requires more effort than they initially thought. Other courses workloads also began increasing, thereby forcing us to even meet and work for hours outside of our schedule.

Since we all had very different strengths while making this project, and there is a difference if you devide the percentages by the amount of input, or the amount of output, it was very difficult to come up with a fair devision of percentages, but finally we agreed on the following:

Koen Klaren	: 27.0	%
Tim Segers	: 26.0	%
Thomas Musters	: 24.0	%
Ky-anh Tran	: 23.0	%