Internship Report

Christopher Chedeau   **July-August 2009**

« Web Development - Guild Recruitment Application
Using Javascript Technology for an Optimal User Experience »

Internship Report

Introduction

Curse was first a website aggregating World of Warcraft interface modifications. It rapidly grew into a MMO (Multiplayer Online Game) network of a dozen of game related websites. The overall network is now being visited by more than 7 million unique visitors a month.

WorldofRaids.com is a website that gives news about what is going on in the World of Warcraft community. In order to provide more services to the player, we decided to launch a Guild Recruitment section on it.

The objective was to use the latest web technologies in order to give the user the best possible experience. My work has been supervised by Whalen Rozelle (project manager) and Hubert Thieblot (Curse CEO) and we were helped by Kevin Van Ness the responsible of WorldofRaids.com

World of Raids - Guild Recruitment

In World of Warcraft, people teams up in structures called guilds. In order to defeat dungeon bosses, they have to bring a balanced composition of players. The overall turnover is pretty high and guilds often look for new players. The project consists in making an interface between players and guilds. You could see that as a Meetic for the gamers.

The principal focus was user experience. There is a lot of data of various nature asked to the user, the existing tools are not great on this point. Our goal is to be able to create and find a guild with the minimum of user interaction. This is achieved using a great deal of Javascript.

Guild Creation

What You See is What You Get

The solution adopted is a What You See Is What You Get form. The administration panel looks exactly the same as the result page; in fact that is exactly the same! This way, the user knows how the page is going to render at all time.

Name Widget

This widget looks to be the simplest one possible but was in fact one of the hardest to add. Here is the technical reason. The field in page is a <h1> and must be converted into an <input> in order to be editable. What I did first was to copy all the CSS properties of the <h1> to the <input>. However I discovered an annoying property: <input> fields have an inner padding!


The brown part is the <input> border and the red part is the inner padding. It cannot be altered by CSS, it depends of the font-size of the text and it is not the same value for every browser. I tried for some times to make a generic conversion using Javascript but it turned out to be very difficult.

Instead, the values for the 3 text fields are hardcoded. This is not an optimal solution, but there is nothing relevant on the internet concerning that issue and the project have to move along.

Realm Widget

In World of Warcraft there are more than 500 realms overall. The widely used method is having a first dropdown for the region and then another one with all the realms of that region. This is really easy to code but could be improved for the user experience.
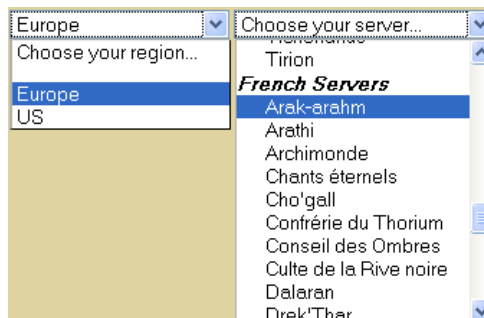
Instead, I opted for an auto-complete solution. All you have to do is type the first few letters of your realm name and it will prompt you the matching realm. It usually takes 3 or 4 characters to have only your realm listed. Instead of having to choose your realm from 2 dropdowns you just have to type a few characters and select it from the list.
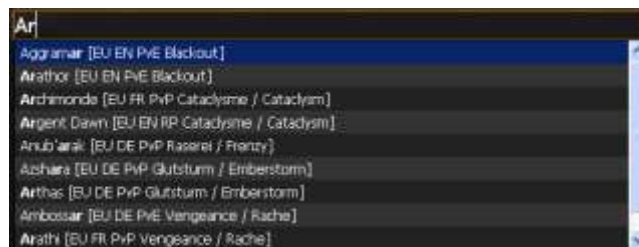
The realm data has been quite hard to gather. In fact there is no realm list available with all the required data so It has to be gathered that list from different official pages with each time a new syntax to parse.

Recruitment Widget

Each class of the game has 3 available specializations. Guilds are either interested in a class or a specialization of a class. This system allows you to choose easily between both. Clicking on the class icon will (un)select all its specializations, and clicking on the specialization will only (un)select it. When selected, the box background matches the in-game color representing the class.
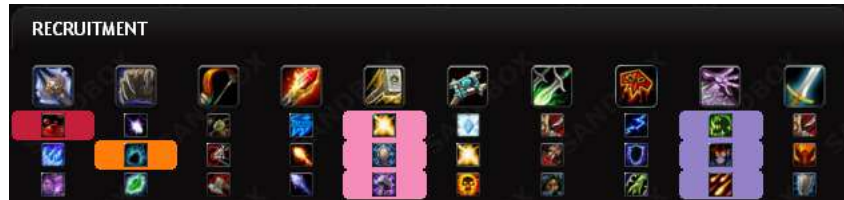
Progression Widget

When you fill the guild name and realm, behind the scene the server is going to the World of Warcraft Armory, the official website where we can view guilds and player, and collects information about the players of the guild. What is great about that is you have all the information filled for you within 15 seconds when you would have spent minutes typing them all.

In the game, there are two sizes of groups (10 and 25 players) for dungeons containing multiple bosses, each with several levels of difficulty. In order to have something easy to maintain with that much data, it has been organized in a configuration file that is easy to fill even for a non programmer.

Calendar Widget

There is no easy way to edit in place the calendar display, instead it is turned into an efficient calendar to fill. You can toggle on and off the days you are raiding by clicking on their name. To alter raid times, you have to press the little -/+ buttons next to the time. Guilds are often raiding at the same times. This is possible to alter raid times of the whole week by clicking on the global signs at the top.

About Us Widget

Always in the What You See Is What You Get spirit, when going into edition mode, a control bar appears and let you do some basic text formatting.
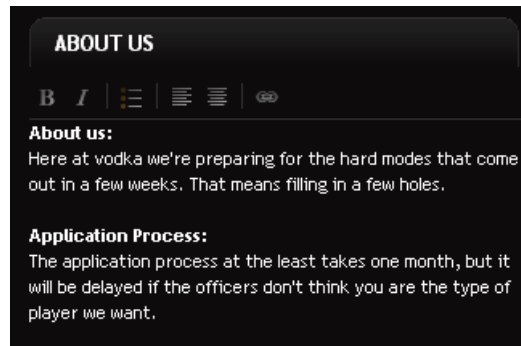
WoWGenuine.com

Having anyone being able to create any guild without any verification is an issue. We don't want anyone to usurp the identity of a guild he doesn't belong to. Also, it is not imaginable to have a manual verification. In order to verify that the user is really part of the guild, we found out that we could use the armory website.

When you logout from the game, your character data is updated in the armory within few minutes. To make sure that the user is genuine, he is asked to keep only few random objects. If the armory updates accordingly, then he has access to the character.

This kind of verification is not only useful to this project but virtually to any World of Warcraft website. Therefore, a new website wowgenuine.com has been developed with a public API to use it.

In order to keep the guild creation process easy, we do not require the guild to be genuine to be published. We want one guild to only appear once in the guild listing. We decided that anyone could publish a guild without being genuine, but in case of conflict, the genuine always wins.

Guild Search

The main challenge with a guild recruitment website is the search ability. The user should find his guild in a minimum time. To accomplish this, we wanted the user to see the guilds evolving as he is tweaking the search filters.

« As You Type » Search

In order to be able to change the results as soon as the user clicks a filter, we had to make extensive use of Javascript. Every guild of the database is sent to the client when he loads the page. This allows not making any server request during the search progress and be able to serve the user within milliseconds.

When a search filter is changed, every guild is being attributed a match value accordingly to the filters already set. Then, all the guilds that have a positive match value are being sorted and the first 10 displayed.

Optimizations

The sorting has been highly optimized to be able to be fast even on older browsers that have not received heavy Javascript optimizations.

There are two things that takes time, the first thing is the HTML rendering. Accessing the DOM is time consuming. Instead of recreating the display every time, the result boxes are kept and only the visible parts that changes are modified. For example if the faction icon is the same as the one previously there, nothing is done.

Also, all the tooltip when you mouse over the icons are generated in real time. The time to compute the tooltip is not perceptible when you use it. However when done about 100 times for all the tooltips of the page it is no more negligible.

Custom URL

Since the application is mostly based on Javascript, the URL has to be handled in the application. We want to store in it the state of the page, which is the combination of the state of every search filter. This is useful when you want to link the page to a friend so he sees the same page as you, and needed when you use the "go back" function of your browser.

The URL should be as short as possible. In order to get the shortest URL, all the filters possible options have their possible option range stored. For example the Class filter can be one of 10 values and the Spec filter one of 3 values. With that list of ranges, we give to each state a number.

> {Class: 1, Spec: 1} ⇔ 0
> {Class: 1, Spec: 2} ⇔ 1
> {Class: 1, Spec: 3} ⇔ 2
> {Class: 2, Spec: 2} ⇔ 3
> … And so on

We put that number at the end of the URL. Instead of storing it with only 0-9 characters, we also use every small and upper case letter. 0 ⇔ 0, …, 9 ⇔ 9, 10 ⇔ a, …, 61 ⇔ Z, 62 ⇔ 10. With this method, we are able to store all the filters in only 18 characters. This is a sample result: *fySA9UJQk4d7mU1DhL*.

Conclusion

The Guild Recruitment project introduced me to the Javascript programming language. I discovered that writing application for the web is both exciting and filled of unexpected difficulties but overall, this gives the user a really great experience.

The application is being polished and will be launched in the following weeks. It is already in beta test, you can access it at http://www.worldofraids.com/recruitment/ with the user "blizzcon2009" and the password "worrecruit2009".

Working in an English environment is not a new thing to me but is always somewhat challenging, especially when it comes to oral. I'm used to write and read English but having to listen and talk is a much harder task. However, I think I made some progress through the course of the internship.

I would like to thank Kevin Van Ness webmaster of WorldofRaids, Whalen Rozelle my project manager and Hubert Thieblot Curse CEO for their trust in me and letting me work with them.