

Curse Internship

Christopher Chedeau

August 2010 to January 2011 - 6 Months



Many aspect of the Web were approached during this Internship such as :

- Community Management and Journalism on SC2Mapster
- Integration of Designs for MMO-Champion and ArenaJunkies
- Development of an Interactive Application for Wowtal
- Automated Versions Analysis for MMO-Champion Database



Curse - 351 California Street - San Francisco, CA 94104, United States

Table des matières

1	Introduction	3
2	Curse	5
2.1	Business Sector	5
2.2	Service	7
3	Contribution	8
3.1	SC2Mapster	8
3.1.1	Start	9
3.1.2	News	9
3.1.3	Contest	10
3.1.4	Moderation	12
3.2	Website Skins	13
3.2.1	MMO-Champion	13
3.2.2	Arena Junkies	15
3.3	Wowtal	16
3.3.1	Layout Reorganisation	16
3.3.2	Additions	17
3.3.3	URL Compression	19
3.4	MMO-Champion Database	22
3.4.1	Patch Download	23
3.4.2	Adapters	25
3.4.3	Changes	25
4	Conclusion	31

Chapitre 1

Introduction

Six years ago, during the beta of World of Warcraft, I met Hubert Thieblot. He wanted to create a website for his guild called Curse while I had the idea of an Addon database for the game. We joined our projects and few months after, Curse.com instantly became the reference. Around the same time World of Warcraft was released and attracted millions of players, becoming the number one-selling game for several years.

The amateur guild website became a company and grew at a fast pace. On my side, I continued my studies (I was 15 at the time). I was still involved in the project and went to the company office one month every year since then. I was officially hired two years ago as a remote employee in order to finance my studies. This 6-month internship is the opportunity to be fully dedicated to this project that was part of my life for all these years.

The company business is based on the Internet that I believe is the platform of today and tomorrow. The company is customer-centric and believes in technology to succeed. And best of all, the core activity is gaming. This is a great fit for me.

My internship was organized in two different phases. I started a website called SC2Mapster three months before the beginning of the internship. It quickly became the biggest place for the Starcraft 2 Custom Maps community. This required me to learn a completely different set of skills as the focus was on advertising, journalism and community management.

During the second part I switched back to a more study-related field which is development. I had the opportunity to work on a lot of different technologies such as design integration with HTML and CSS, interactive application with Javascript and finally web back-end using Python and Django.

I would like to thanks

- **Hubert Thieblot**, the CEO of Curse, for believing in me. Thanks a lot for the job and the internship.
- **Fabien Bonte**, the lead of MMO-Champion, for the great cooperation on your website.
- **Kevin Carlino**, SC2Mapster co-admin, for all the hard work you did to make it a successful website.
- **Bryan McLemore**, my project manager, for giving me the right directions.

Chapitre 2

Curse

2.1 Business Sector

Curse is the largest network of websites focused on Massively Multiplayer Online Games. More than 450 million page views are done by 12 millions unique visitors every month. It provides a wide range of services to gamers.



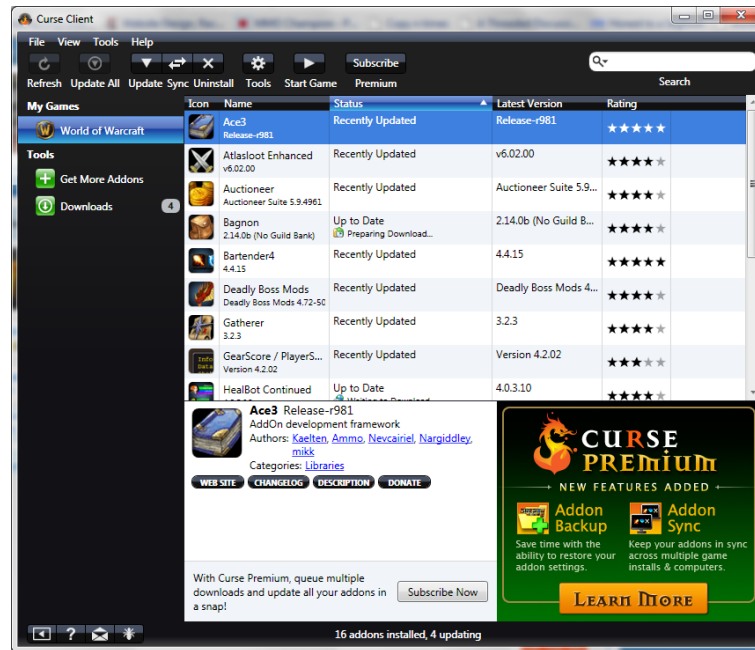
World of Warcraft, the game by Blizzard Entertainment, is responsible for 80% of Curse traffic.

Curse.com is the main website of the network. You can think of it as a World of Warcraft app-store. It provides a wide variety of addons for the game. The gamers can use a wide range of

MMO-Champion provides fresh news about the game and have active forums where gamers communicate.

WowStead is a guild hosting platform. It provides all the required tools for guilds such as a full featured-website including a raid tracker, forums, blogs and instant messaging servers.

Curse Client is a desktop application that makes installation and upgrade of the addons extremely easy. A premium service that costs \$3/month gives an automatic upgrade of all the addons among other services.



Wowpedia is a Wikipedia for World of Warcraft.

ArenaJunkies is a community website that focuses on the Player vs Player aspect of the game.

Beside World of Warcraft, Curse runs websites for many other games. **MinecraftForums**, **MinecraftWiki** are the official websites of Minecraft. **Zybez** is a database for the free to play game Runescape. **AionSource**, **AionArmory** are a community website and its database counterpart for the game Aion. There are also websites for the Blizzard Game Starcraft 2 such as **SC2Mapster** for the map makers and **SC2Replayed** to grab the latest pro replays. And more, there is a total of 33 websites owned by Curse.

2.2 Service

The company is organized in three divisions. There is an important number of people dedicated to do the community management and content creation. Many of them are remote employees. Then, the technology team, based in Huntsville is dedicated to develop all the applications behind the websites. Finally there is a team in San Francisco that is responsible to generate revenue.



Project Management (3 people) Manage projects, lead product development design and overview customer service.

Content and Community (23 people) There are many websites under Curse responsibility. The content team is providing the latest news about the covered games. The other mission is to manage the community that lives around those websites. They have to moderate the forums, organize contests, and so on ...

Technology (10 people) Websites are powered by many different applications. Curse uses and administrate external ones such as VBulletin or Invision Power Boards. The technology team also develops internal framework such as Cobalt (curse.com, wowace.com), CurseForge (curseforge.com, wowace.com, sc2mapster.com) and the Curse Client, a desktop application.

Monetization and Sales (10 people) There are 450 million page impressions per month on the network. The team is charged of selling advertising placement at the best price. They also manage the Premium services of the Curse Client and WowStead.

Finance (2 people) Handles Finance activities, Human Relations, Legal and Business operation.

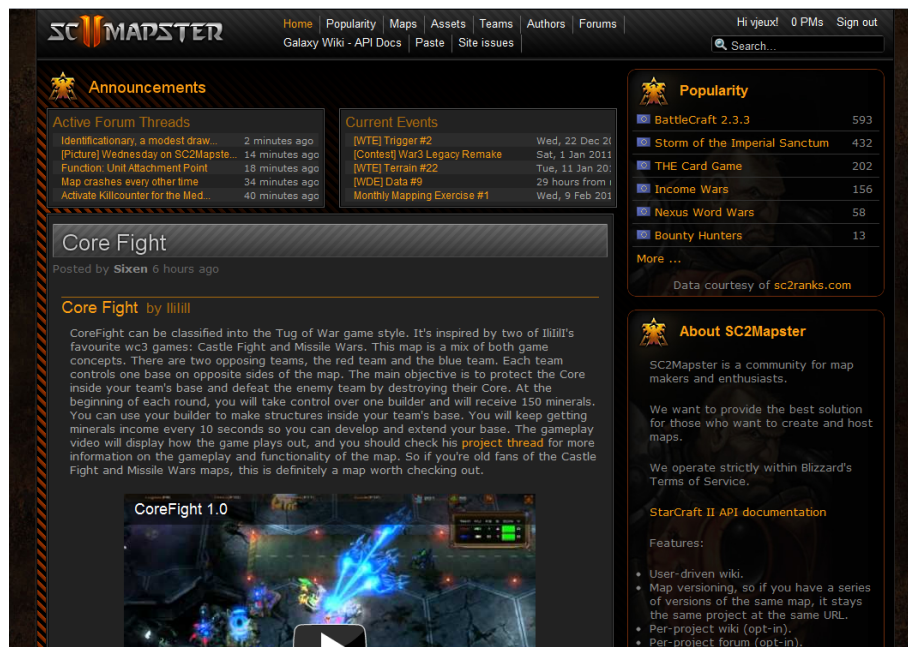
During the first part of the internship I was a member of the Community team while running SC2Mapster. Then I moved to the Technology team.

Chapitre 3

Contribution

3.1 SC2Mapster

Let's get back three months before the beginning of my internship. Blizzard just released the private beta of the long waited real-time strategy (RTS) game Starcraft 2, a remake of their first extremely-successful game. I wanted to run a website for this promising game and started SC2Mapster.com. It is focused around the Map Editor of the game.



3.1.1 Start

I spent countless hours working on custom maps for Warcraft 3, the other RTS franchise from Blizzard. Therefore I decided to run SC2Mapster. I did not start the website from nothing, we adapted a custom web framework called CurseForge that we use for the addon developers. It has built-in support for file hosting, forums, announcements, wiki, etc... Everything needed!

Great, now it's time to get people in. At the time of the launch there were three kinds of competitors : important Warcraft 3 websites about custom maps such as HiveWorkshop¹, WC3Campaigns² or Nibbits³, huge portals about Starcraft 2 with a mapping part like TeamLiquid⁴ and new sharks like us.

The strategy that I use was to behave like a user. The map editor was not yet released but we could analyze the already existing maps. During this period I started contacting the few users that were hacking the maps and told them to come on the IRC. At the same time I started an effort to document all the parts of the editor. In order to get traffic from outside, I created the first custom map that was not just a basic modification of existing maps : Mapster TD⁵. It has been linked on various non-mapping related websites and the video was viewed 35 000 times.

3.1.2 News

In the early days of Starcraft 2, it was really difficult to know what happened in the modding community. Everyone was working on his own and did not communicate with each other. I spent a lot of time seeking for websites and people that were acting in the area. Since I found some cool projects, I started listing those in the front page of SC2Mapster.

Quickly, people started to visit the website as they wanted to know about the Starcraft 2 map making. At the beginning the majority of links were sending people outside of SC2Mapster. Interestingly, people stucked to our website. They started contributing on the forums thanks to the news I was writing.

As it shows great results, I decided to start writing news at a daily rate. This was quite a challenge as I never had any experience in journalism. At first this was hard. I did not really know where to find good source of

1. <http://www.hiveworkshop.com/>

2. <http://www.wc3c.net/>

3. <http://sc2.nibbits.com/>

4. <http://www.teamliquid.net/>

5. <http://www.sc2mapster.com/maps/mapster-td/>

information. There were many interesting projects going on but they rarely provided material that was good to show. I started contacting people and ask them to make content that I could news. In 6 months, not a single person did not answer me or did not wanted to do it, this was really impressive !

I found that the best way to show off maps was video. The other options are text or screenshots. Text is painful to read and does not represent well a map. Screenshots are better but it is really hard to show liveliness with it. It is also painful to browse more than 5 screenshots. I decided to use video as primary support, if a project does not have a video then it will not be newsed. About a month after the adoption of that guideline, a lot of project started to do videos. It substantially improved the communication quality of a lot of projects.

3.1.3 Contest

A contest is an effective way to stimulate a community. The Starcraft 2 map editor being powerful, it gives wide range of areas to explore. When a contest takes place, this is the opportunity to make everyone focus on the same objective for a given period of time. This is also a motivator since the entries will be rated and a winner elected.

Contests were organized at a rate of two per month. One week to do the entry, one week to vote. As you can see, all the thirteen contests were unique.

- **World of Warcraft Boss Contest** September 21st, 2010
- **Epic Battle Scene Contest** September 2nd, 2010
- **Blizzard Custom Map Contest** August 13th, 2010
- **User Interface Contest** August 4th, 2010
- **Minigame Contest** July 21th, 2010
- **Creature Contest** June 29th, 2010
- **Wallpaper Contest** June 17th, 2010
- **Spells Contest** June 9th, 2010
- **3D Modeling Contest** May 26th, 2010
- **Melee AI Contest** May 25th, 2010
- **Cut Scene Contest** May 14th, 2010
- **Clock Contest** May 2nd, 2010
- **Inventory Contest** April 11th 2010

How to make a good contest

This is hard to organize a good contest.

The most important thing is to define the objective : stimulate the community. The community as a whole needs to participate. The rules of the contest should welcome both the newcomer and the expert. The challenge lies in the huge difference in skill between both, we want to make sure that everyone has a chance to win.

The first step toward this goal is to limit the time of the contest. Giving a week is enough for people to make interesting things, but not enough for a polished product. Therefore all the entries will be prototypes. The concept submission will be remain important factor. A week is also short enough so that people will start working right away, and not be tempted to do it later on, and then forget.

Then the subject is important. It must be balanced between specificity and openness. Each contest should have a specific theme so people have the feeling that they are competing on the same field. But it should give the opportunity to innovate. You want to vary the themes of your contests and make them cover all the features of the domain.

Since not everyone will participate, we make people vote for the best entry. This voting phase will show all the entries to the community. The results will be biased as people who vote will probably judge only with the provided videos and not test the map. But this is not important as the goal is to get people involved !

Incentives. I found out that people would gladly enter the contest even if there was no real prize. Being highlighted in the front-page of sc2mapster is motivating enough. If we were to put cash prizes, people would be less likely to do it for fun but try instead to abuse the system to win.

Finally, listen to the community. A lot of people came to me with contest ideas. However take care, most of them will not be good contests. Make sure you adapt them to fit all the rules I highlighted and eventually they will be good to go.

3.1.4 Moderation

The site got bigger. At one message per minute at peek hours, it was no longer able to review all the messages that were posted. Therefore it became difficult to moderate the forums.

We made a call for moderators and we were really surprised that 40 people applied. It was a tough decision to chose them. In the end one moderator was assigned for each forum, and three for the most active one. Using the activity count and their preferences we picked 13 out of all the applicants.

I had great faith in the moderators we took and wanted to give them the ability to conduct a project of their choice. It would be advertised on the front-page and labeled as an official SC2Mapster project. However this failed. This was more perceived as a constraint instead of an opportunity. It also confused them as to what were their role.

Another mistake we made was a lack of communication. We did not force the moderators to go into our private IRC channel. Therefore we were not able to create a group feeling. This was amplified by the timezone differences : the Europeans rarely talked to the Americans.

Even if I wasn't a really good manager, they did an awesome job at moderating the forums.

3.2 Website Skins

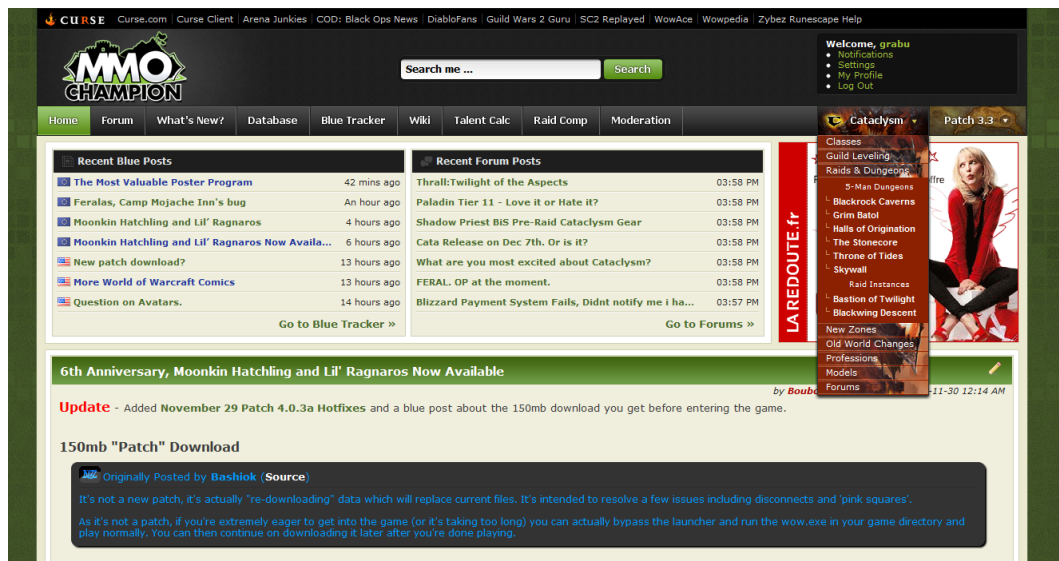
3.2.1 MMO-Champion

MMO-Champion has the same theme since its launch, 4 years ago. Brandon Ferrua, Curse lead designer, made a new design for MMO-Champion. Two years ago, I implemented again the MMO-Champion layout from scratch with performance in mind. Therefore I was naturally given the task.

You can see the old design in the following picture.



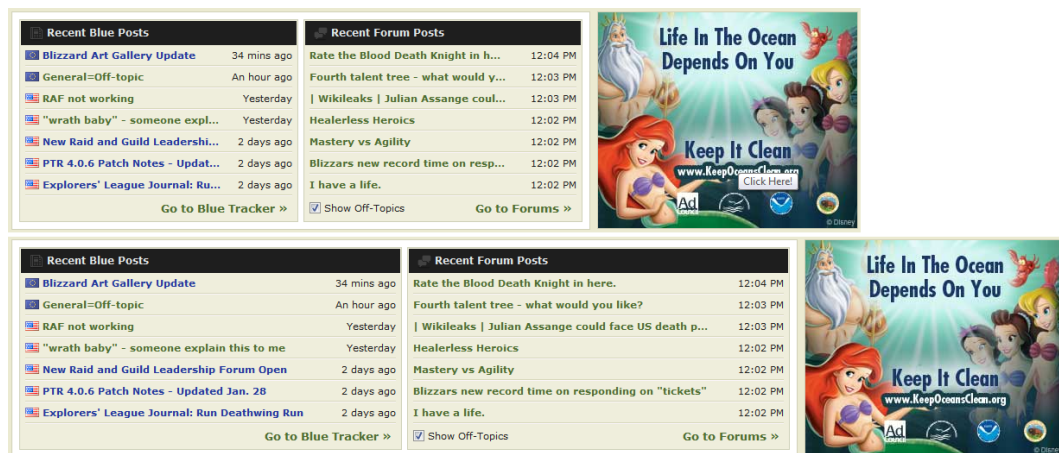
The new design looks more professional as it uses less flashy colors. The main objective though was a layout reorganization. The side-bar has been removed, the previous content is now displayed as a drop-down menu. This gives news the full width of the page. The Recent Activity area is better integrated to provide a better visibility.



The main page was quite straightforward to integrate. The designer gave me a design for 4 different pages. However I had to redesigned all the other pages. I had to reuse a lot the parts that were given to fill the non-designed parts.

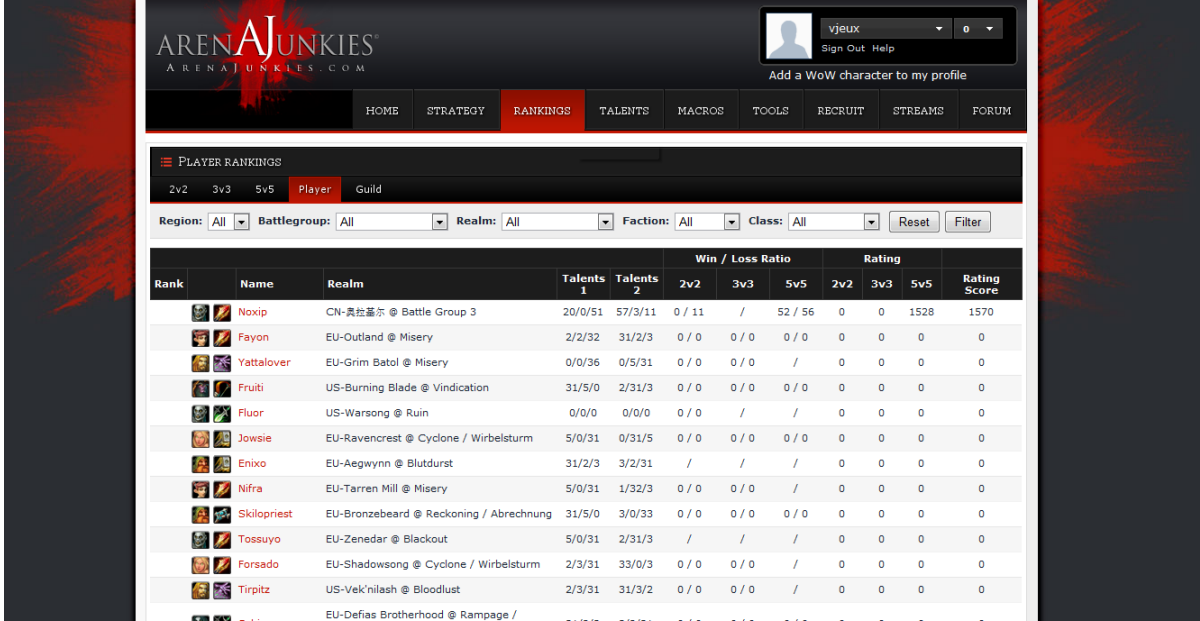
Recent Activity Bar

The part where I struggled the most is the Recent Activity Bar. The design needed to be with a flexible-width. CSS styling was not designed with this objective in mind. The two boxes containing the recent activities must extend horizontally but the advertising needs to keep the same width. Also, the text must not extend beyond the available space.



3.2.2 Arena Junkies

Since I successfully integrated MMO-Champion newest design, later on I was given the same thing to do with Arena Junkies. It was very similar.



The screenshot shows the Arena Junkies website interface. At the top, there's a navigation bar with links: HOME, STRATEGY, RANKINGS (active), TALENTS, MACROS, TOOLS, RECRUIT, STREAMS, and FORUM. Below this is a 'PLAYER RANKINGS' section with tabs for 2v2, 3v3, 5v5, Player (active), and Guild. There are filters for Region, Battlegroup, Realm, Faction, and Class, along with 'Reset' and 'Filter' buttons. The main table displays player rankings with columns for Rank, Name, Realm, Talents 1, Talents 2, Win / Loss Ratio (2v2, 3v3, 5v5), Rating (2v2, 3v3, 5v5), and Rating Score.

Rank	Name	Realm	Talents 1	Talents 2	Win / Loss Ratio			Rating			Rating Score
					2v2	3v3	5v5	2v2	3v3	5v5	
1	Noxip	CN-奥拉基尔 @ Battle Group 3	20/0/51	57/3/11	0 / 11	/	52 / 56	0	0	1528	1570
2	Fayon	EU-Outland @ Misery	2/2/32	31/2/3	0 / 0	0 / 0	0 / 0	0	0	0	0
3	Yattalover	EU-Grim Batol @ Misery	0/0/36	0/5/31	0 / 0	0 / 0	/	0	0	0	0
4	Fruiti	US-Burning Blade @ Vindication	31/5/0	2/31/3	0 / 0	0 / 0	0 / 0	0	0	0	0
5	Fluor	US-Warsong @ Ruin	0/0/0	0/0/0	0 / 0	/	/	0	0	0	0
6	Jowsie	EU-Ravencrest @ Cyclone / Wirbelsturm	5/0/31	0/31/5	0 / 0	0 / 0	0 / 0	0	0	0	0
7	Enixo	EU-Aegwynn @ Blutdurst	31/2/3	3/2/31	/	/	/	0	0	0	0
8	Nifra	EU-Tarren Mill @ Misery	5/0/31	1/32/3	0 / 0	0 / 0	/	0	0	0	0
9	Skilopriest	EU-Bronzebeard @ Reckoning / Abrechnung	31/5/0	3/0/33	0 / 0	0 / 0	0 / 0	0	0	0	0
10	Tossuyo	EU-Zenedar @ Blackout	5/0/31	2/31/3	/	/	/	0	0	0	0
11	Forsado	EU-Shadowsong @ Cyclone / Wirbelsturm	2/3/31	33/0/3	0 / 0	0 / 0	/	0	0	0	0
12	Tirpitz	US-Vek'nilash @ Bloodlust	2/3/31	31/3/2	0 / 0	0 / 0	/	0	0	0	0
13	Schisma	EU-Defias Brotherhood @ Rampage /	31/2/3	3/2/31	0 / 0	0 / 0	0 / 0	0	0	0	0

3.3 Wowtal

In World of Warcraft you can customize your character using the talent system. You have to select from a wide variety of bonuses the ones you want. This is a really important decision as it is costly to change. A tool called Wowtal has been written by Nathan Adams and Jerome Leclanche to simulate the talent system in the browser.

With the release of the expansion of World of Warcraft, the tool requires to be upgraded. First of all, the user interface needs to be improved as tools from the concurrent did it better. It consists mostly in cosmetic changes and organizing the space more intuitively.

Then, the core work is to implement the glyph selection. Along with the talent system, Blizzard added a new way to customize the character with its own mechanism.

The result of this specialization is given through a URL such as <http://wowtal.com/#k=YUej0g-J.aei.hunter.adcx12>. In order to keep it small, a data compressor is required.

The day the upgraded tool was launched, it has been loaded 100 million times.

3.3.1 Layout Reorganisation



The sidebar is not working well because it breaks locality. Trees informations (name, points spent, reset) should be near the tree and not regrouped somewhere else. I move those elements at the top of the trees. But now the sidebar feels empty. I created a new horizontal bar to hold the information

and it became a lot more user-friendly. At the same time, we reduced the width requirement allowing it to run better in smaller monitors.



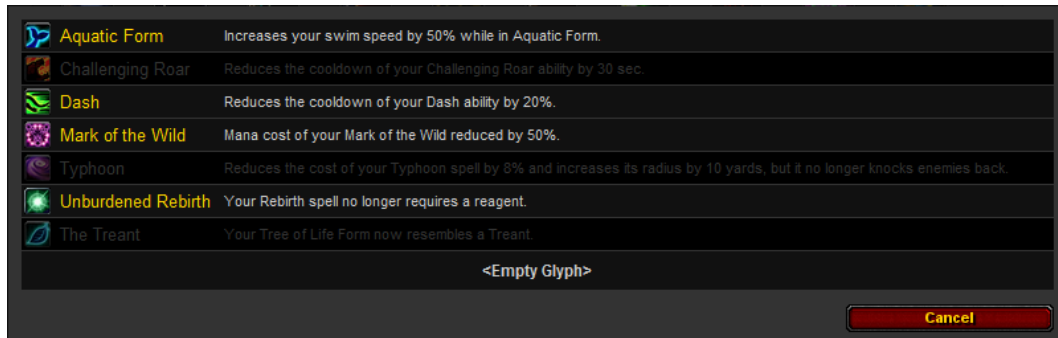
3.3.2 Additions

Glyphs

Blizzard added a new way to customize your character : Glyphs. You have to chose three glyphs among all the existing ones. They are sorted into three distinct categories : Prime, Major and Minor. Since it there are three columns, we decided to layout them like the talent trees.



When you click on one, a list of all the available glyphs appears as an overlay. Already selected glyphs are disabled.



Summary

In order to help beginners, Blizzard added a summary of each tree including an icon, description and the important abilities. I reproduced it in the application.



Use it ingame

It is annoying and error prone to translate the build in-game. Maldivia, a member of the MMO-Champion IRC channel, found a way to generate a

line that, when copy & pasted in-game, reproduce the build you've made. The challenge was to make it fit under 255 characters, the maximum length you can enter in the chat box.

```
/run t,p,a={1,13,23,32,53,62,72,81,91,102,112,122,131,142,151,161,171,182,
193,201,2,3,22,43,61,}SetPreviewPrimaryTalentTree(t[1],GetActiveTalentGroup())
for i=1,#t do a=t[i]if a<9 then p=a else AddPreviewTalentPoints(p,floor(a/10),
a%10)end end
```

And the version with proper indentation and comments :

```

/run t, p, a = { // Talent definition
    1, 13,23,32,53,62,72,81,91,102,112,122,131,142,151,161,171,182,193,201,
    2,
    3, 22,43,61,
}
// Open the talent panel
SetPreviewPrimaryTalentTree(t[1], GetActiveTalentGroup())
for i = 1, #t do
    a = t[i]
    if a < 9 then // If one digit, then that's the tree number
        p = a
    else // The first digit is the column, the second is the row
        AddPreviewTalentPoints(p, floor(a / 10), a % 10)
    end
end
end

```

3.3.3 URL Compression

People want to share their build on the forums, instant messaging, etc ... You do it by copy & pasting the URL of the web page you are watching. It must contain all the information able to reconstruct your points. The first talent calculators saved it using one character per point which looked something like that.

- http://wowtal.com/#hunter.3230523013052133323110123000000000
00000000000000125155000000000000

This is a 64 character string, which is really long. We can compress it. One character is being used for each talent. Each talent cannot have more than 6 different values (0..5) but a single character can hold a lot more (a..z, A..Z, 0..9). We would like to encode as much data as possible in it. To do that, we can treat our build as a number in base 6. We first decode it into a number the computer can understand and then encode it into a base 64 string.

```
// Decode the base6 string into a number the computer can understand
alphabet = "012345"
number = 0
foreach (char in base6) {
    number *= 6
    number += alphabet.pos(char)
}

// Encode it into a base64 string
alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-_"
base64 = ""
while (number != 0) {
    base64 = base64 . alphabet[number % 64]
    number = number / 64
}
```

The decoding process is trivial, you just swap the alphabet and bases : decode a base64 string and encode it back into a base6 string.

There is a total of 64 talents, with 6 possibilities per talent, encoded into a base 64 string. The number of required characters to encode all the possible builds is $\lceil \log_{64} 6^{64} \rceil = 28$. This is better than 64 but can still be improved.

As you can see, we multiply the number of possible options for each rank together. It grows extremely quickly. We should exploit the fact that all the talents do not have 5 ranks. The number of required characters is now $\lceil \log_{64} (\prod_i^i \text{ranks}_i + 1) \rceil = 17$.

We have to reflect this technique in our code. Instead of decoding a base 6 string, we now decode a string with a base that varies for each character.

```
i = 0
number = 0
foreach (char in input) {
    number *= ranks[i] + 1 // Use a different base for every character
    number += alphabet.pos(input)
    i += 1
}
```

This is the optimal solution if the talent repartition is uniform. But it is not. You have to chose a primary tree where you will spend a minimum of 31 points. The 10 remaining points can be spent in any trees. Another rule impose you to spend 5 points in a row before you can spend any point on the next row. This means that only the first two rows of your non-primary tree can be filled. This reduces the number of eligible talents from 64 to 32.

We have to encode a new information : the primary tree (3 possibilities). The character count required to encode the talents is down to $\lceil \log_{64} (3 \prod_i^i \text{ranks}_i^{\text{tree}} + 1) \rceil = 10$.

Applying this technique on **talents**, **build number** and **glyphs**, we generate a URL that is small enough.

- `http ://wowntal.com/#twABis181.adl.hunter.-deEJh`

3.4 MMO-Champion Database

Fabien Bonte runs MMO-Champion⁶, a website that provides latest news about the game World of Warcraft. Blizzard is releasing patches at a pace of about 2 per week. There are millions of players eager to know how their game is evolving. He makes most of his traffic by data mining the patches and telling what has changed. This is an area with a lot of concurrence as it generates high volumes of page views.

Since he is not a developer he relies on external people to do content extraction. This is problematic as they have to be available when a patch comes out, even in the middle of the night, to execute the extraction. MMO-Champion side-project is a World of Warcraft database⁷ that displays every item, spell, etc ...

My objective is to use it as a framework to automatically generate differences between two version of the game. This output is going to be analyzed by Fabien Bonte that will post important information on the website.

Unfortunately, Jerome Leclanche, who wrote the database, left the company right before I started the project. I had to get familiar with his code base on my own.

6. <http://www.mmo-champion.com/>

7. <http://db.mmo-champion.com/>

3.4.1 Patch Download

First of all, we need to obtain the latest version of the game. The most reliable way to do this is to analyze how the World of Warcraft Launcher does it. I analyzed network packets with Wireshark⁸ and reverse engineered its communication protocol.

The launcher first sends a HTTP request to the patch server⁹ with the following information.

- **Realm** : enUS, enGB or public-test
- **Tool Version** : 2086, 2105, 2121
- **WoW Version** : 1 (live) or 2 (public-test)

The patch server, through various XML transactions and links, leads us to a custom torrent file listing of all the available files. A custom implementation of bencode algorithm¹⁰ has been written to read the torrent.

Tool and WoW versions are sometimes changing, an utility has been written to be able to find them easily. It goes over all the possible values and stores the working ones.

Guess Build

We can download the patch like everyone else, however we want to get it before anyone else ! We remarked that the links used to download the patch follow a pattern :

- [http://ak.worldofwarcraft.com.\[...\]/wow-update-enGB-13449.MPQ](http://ak.worldofwarcraft.com.[...]/wow-update-enGB-13449.MPQ)¹¹
- [http://ak.worldofwarcraft.com.\[...\]/wow-update-enGB-13529.MPQ](http://ak.worldofwarcraft.com.[...]/wow-update-enGB-13529.MPQ)
- [http://ak.worldofwarcraft.com.\[...\]/wow-update-enGB-13561.MPQ](http://ak.worldofwarcraft.com.[...]/wow-update-enGB-13561.MPQ)

You probably have the same idea as we did : we can query the next 100 ids and see if any match. Experience showed that Blizzard uploads the patch around 3 or 4 hours before deploying it on the patch server.

We just have to be careful not abusing of this system because there is a flood limit. Once you reach it you are banned for 24 hours.

8. <http://www.wireshark.org/>

9. <http://<realm>.patch.battle.net:1119/patch>

10. <http://wiki.theory.org/BitTorrentSpecification>

11. <http://ak.worldofwarcraft.com.edgesuite.net/wow-pod/ptr/direct/base.12911.4.0.0/Data/enUS/wow-update-enGB-13449.MPQ>

MPQ Management

As you can see, a patch is a MPQ^{12 13} file. This is an archive containing files. The archive format is evolving, and it is vital to rely on a maintained library. The project was using libmpq¹⁴ but it does not handle the newest MPQ files. Therefore, I moved to StormLib¹⁵ by Ladislav Zezula. It was started 10 years ago and is still being actively worked on.

StormLib however did not have any python bindings¹⁶. Instead of using the legacy C++ API, I decided to make a user-friendly API that follows the python standards :

MPQ(path) : mpq Open a MPQ archive
mpq.patch(path) : files Append patch MPQ archives
mpq.list(mask, ...) : files Return all the files that matches the mask
mpq.has(path) : boolean Test if a file is in the MPQ
file.read() : string Return the content of the file
file.extract(path) Extract a file to the specified path

In the process of doing this bindings, I helped Ladislav Zezula fix some minor issues in the library.

DBC Extractor

The information that is useful to us is stored as a database inside the game client. Each table being stored as a DBC¹⁷ (DataBase Client-side) file. This is a binary file composed of consecutive records. It contains everything the client needs to play the game.

Thanks to the MPQ infrastructure we set-up, it is easy to extract those files.

```
from storm import MPQ
mpq = MPQ('locale-enGB.MPQ').patch('wow-update-enGB-*.MPQ')
for file in mpq.list('*.dbc', '*.db2'):
    file.extract()
```

A command line tool was also developed to allows Fabien Bonte, who is not a developer, to manipulate MPQ files.

```
mpq locale-enGB.MPQ --patch 'wow-update-enGB-*.MPQ' extract '*.dbc', '*.db2'
```

12. <http://www.zezula.net/en/mpq/mpqformat.html>

13. http://wiki.devklog.net/index.php?title=The_MoPaQ_Archive_Format

14. <https://libmpq.org/>

15. <http://www.zezula.net/en/mpq/stormlib.html>

16. <https://github.com/vjeux/pyStormLib>

17. <http://www.madx.dk/wowdev/wiki/index.php?title=DBC>

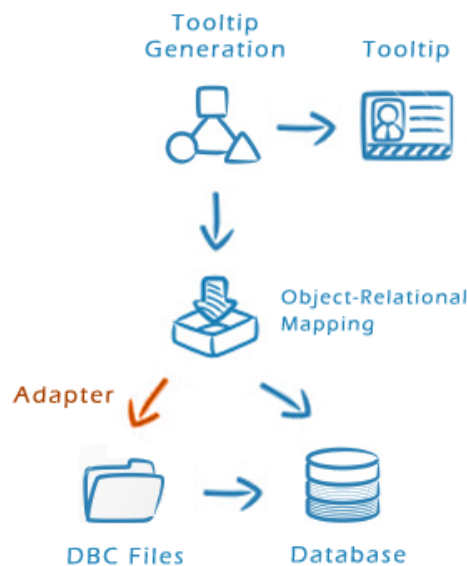
3.4.2 Adapters

In the current architecture, DBC files are parsed and added to the database. With the time, Blizzard changed the DBC structure. Instead of updating the database structure, an import script was written to transform the data fit the old scheme. In the end, the project database and DBCs diverged.

To retrieve an object from the database, we use an Object-Relational Mapping¹⁸ (ORM), the one provided by Django^{19 20}. However it does not support multiple versions of a database. Therefore we cannot use any of our existing code on more than one version at a time.

I made a modification to the ORM so that it reads data from DBC files instead of the database. Real issue is to re-apply all the changes required to convert the DBC structure to the database model. The import script is about 2000 lines long and hardly re-usable.

As this is a design issue, it requires to rewrite an important part of the project in order to be able to read data from both DBCs and database.



3.4.3 Changes

We are able to get the latest version of the game, extract data and read it. Next step is to compare both versions and display the two versions side-by-side and highlight the differences.

18. http://en.wikipedia.org/wiki/Objectrelational_mapping

19. <http://www.djangoproject.com/>

20. <http://docs.djangoproject.com/en/dev/ref/models/instances/>

Detect Changes

We extract all the elements of two tables, sort them by ids and traverse them in parallel. If an element doesn't exist on one side, it was either added or deleted. When both exist, we analyze them field by field.

Text Change

Since values can be of many forms : string, numbers, bitmask, etc ... they are first converted into a text representation.

I used a script called `simplifiediff`²¹ by Paul Butler that compares two lists of tokens and marks the ones that are different. It recursively extracts the longest common substrings²² and calls itself back on the remaining parts. Once there are no common substrings, the remaining tokens are flagged as changed. Let us look at an example :

Deals **90.74** to **98.3 Holy** damage. Deals **146.09** to **158.26 Shadow** damage.

At first I used a character as a token, but it gives inappropriate results. From this example we can see that we want to diff both strings word by word instead of character by character.

Deals **90.74** to **98.3 Holy** damage. Deals **146.09** to **158.26 Shadow** damage.

HTML Change

In order to mark a piece of information as changed, we wrap it within a HTML tag. While it works fine for plain text, when input is a HTML fragment, the insertion of this tag may result in an invalid HTML as seen in those examples :

```
<div <ins>class="item"></ins>
<ins><div></ins> Text <ins><div></ins>
```

This was a tough issue to deal with but after many trials and errors I eventually found a working procedure.

- Extract tokens from Text Nodes of the HTML tree
- Compare the two lists of tokens
- Rebuild the HTML tree from the old one marking the modified tokens

Let us explain the procedure with simple example.

21. <https://github.com/paulgb/simplifiediff/>

22. http://en.wikipedia.org/wiki/Longest_common_substring_problem/

```

<item>                                <item>
  <line>+341 Intellect</line>          <line>+321 Intellect</line>
  <line>Requires Level 80</line>       <line>Red Socket</line>
</item>                               <line>Requires Level 85</line>
                                      </item>

```

A key step here is to use knowledge of the input type. Since it is a valid HTML document, instead of considering it as a plain list of token we can see it as a tree. It contains two types of nodes. **Tags Nodes** are structural elements whereas **Text Nodes** that stores content. We want to preserve the structure of the input but highlight content change. Therefore we will extract tokens from **Text Nodes**.

```

"+341", "Intellect", "Requires",    "+321", "Intellect", "Red", "Socket",
"Level", "80"                      "Requires", "Level", "85"

```

Using the Diff utility we mark the differences.

```

"+341", "Intellect", "Requires",    "+321", "Intellect", "Red", "Socket",
"Level", "80"                      "Requires", "Level", "85"

```

We go over the two inputs and rebuild a HTML output. **Text Nodes** are tokenized and tokens flagged as different are safely wrapped with a tag. **Tags Nodes** are left unmodified.

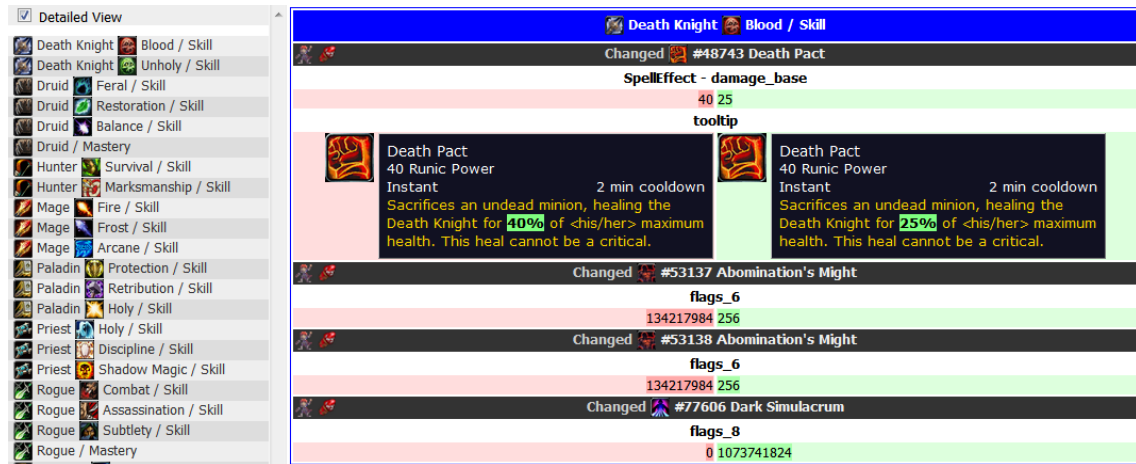
```

<item>                                <item>
  <line>[+341] Intellect</line>        <line>[+321] Intellect</line>
  <line>Requires Level [80]</line>     <line>[Red Socket]</line>
</item>                               <line>Requires Level [85]</line>
                                      </item>

```

HTML Output

The changes are analyzed by Fabien Bonte. He extracts important changes and format them into a patch note. It is therefore important to find a way to present the important amount of data (there are often thousands of changes in a patch) in a readable way for a human.



The output is saved in a HTML web page. The standard field to field changes have been enhanced with many features in order to make the processing easier.

Category Similar objects are grouped into a multi-level category system. This gives an essential context to the object.

Menu On the left side of the screen, there is a sorted list of all the categories. You can click on them to see the associated changes on the right. This gives a fast vision of what changed and makes navigation easy.

URL Each category and object is given a specific URL to improve communication.

Related Links Links related to the object are given. It gives extra informations when needed.

BBCode When clicking on any object referenced in the tooltip, a popup appears with a copy & pasteable forum markup (called BBCode).

```
[url=http://db.mmo-champion.com/item/64401/standard-of-unity/]
[color=#0080ff]Standard of Unity[/color][url]
```

Detailed View In order to have a more compact view, you can uncheck the "Detailed View" option and it will only displays tooltip changes. This makes browsing all the changes faster when you want a global vision.

Progressive Loading Interaction code, required to do all the previously explained features, is written in a way that allows it be executed even if the output is not complete. This makes possible to use the partial output even before the process completes, which can takes several minutes.

Patch Note Output

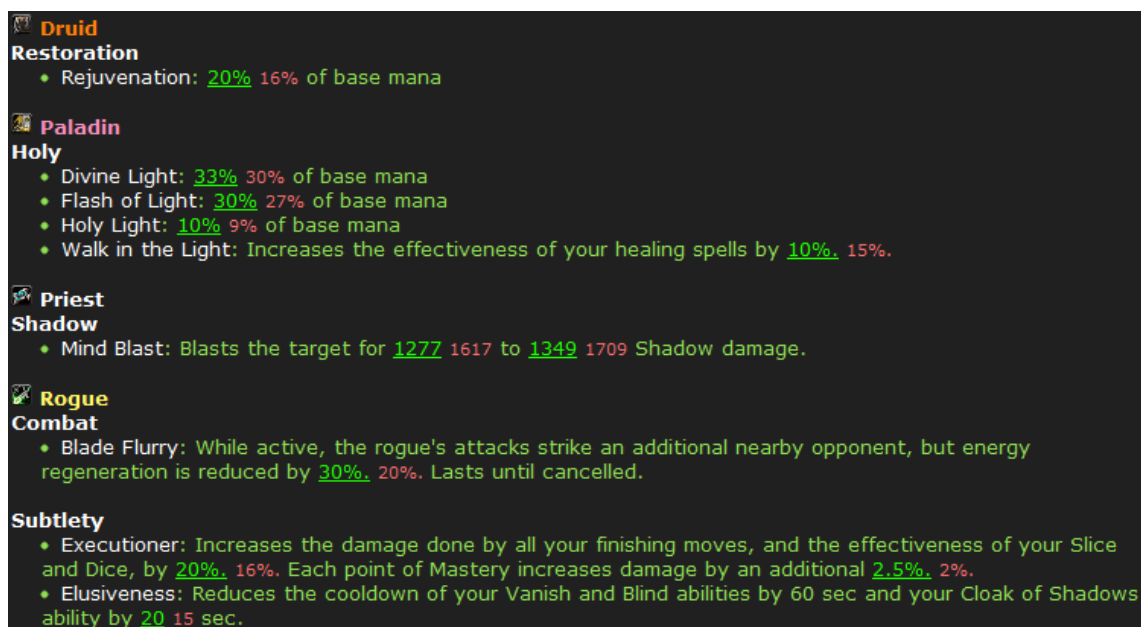
The business of patch notes is very important as millions of players want to know how the game is going to change. The first website to get the note up is likely to be linked everywhere and receive all the traffic.

A manual patch note takes about 2 or 3 hours to write. Knowing that, our concurrent Wowhead²³ set up an automatic patch note generation. In order to be competitive, we have to do it too. It will only be published while the manual patch note is being written.

A challenge here is to be able to show the differences in a textual and compact way. We first eliminate all the changes that does not affect the tooltip as they have great chances not to be important.

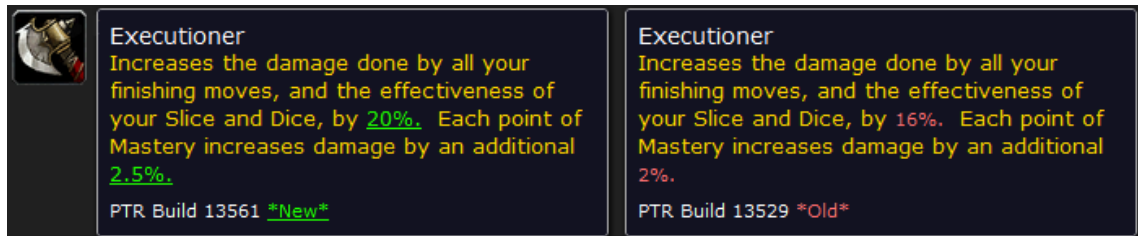
Often the change is only affecting a small portion of the tooltip. Hopefully tooltips are organized into many different sections (name, mana cost, description, etc...). We can delimit sections in the token list by inserting a special token at the end of each section. Only sections that are modified are outputted.

The last modification is to combine both versions into a single line. Additions are underlined and written in green whereas deletion are smaller and red.



23. <http://www.wowhead.com/>

When you put your mouse over the object name, both tooltips are being shown side-by-side.



Extensions

At first, the objective was to generate the spell differences. It turns out that the code was easily generalized to other data types. You need to specify how to generate a tooltip and assign a category for every object. The following data types are handled :

- Spell
- Item
- Achievement
- Quest
- Creature

After four years doing manual patch notes, Fabien Bonte is now getting proper tools to work with.

Chapitre 4

Conclusion

During the first part of the internship, I did not do any development but relational. I launched the website SC2Mapster from scratch. I face many issues that I would not have thought of. For example it is easy to announce a new feature when you already have the user base, you just have to talk about it all over your website. Releasing a new website is tougher. It is not quite possible to go where targeted people are (on concurrent websites) and advertise there.

In order to attract people I started writing daily news about the domain. However, I had no journalism background at all. I had to setup an editorial line and build reliable sources of information. The flow of information is not regular : sometimes there is nothing important being released. it is important to plan one or more days ahead to be able to identify those times and have the time to do a deeper content search.

SC2Mapster got bigger and bigger and I had to manage the community. I organized contests every two weeks to keep it active. This was a win-win situation as it also provided news content. But it also comes with downsides. Some people started to be disrespectful on the forums, bots kept posting spam messages and so on. To overcome this situation, moderators were chosen. I was now discovering how to be a good manager.

It introduced me to a completely new domain but I wanted to switch the focus of the internship back to development. There was a developer spot to fill at MMO-Champion and I took it.

The first mission was to integrate a new design, it was the occasion for me to perfect my HTML and CSS skills. Then I improved Wowtal, a complex Javascript application. While those tasks did not represent a huge challenge, they had to be done carefully as MMO-Champion is among the 1000 biggest websites. A lot of testing was made, ensuring that it would not break on unusual configurations.

Right after, I have been given the lead on the World of Warcraft Database part of MMO-Champion, as its developer left the company. It is a software composed of 10 000 lines of Python, written with the framework Django. I did not know either one, which made the task difficult.

My goal was to automatically generate a change log for a given World of Warcraft Patch. To achieve this objective, I made a robust system that downloads patches as soon as they land, spent time working on a function that compares two strings and used business knowledge to present the data in a useful way.

To conclude, this internship was a great experience. I was able to get an insight of what would a start-up creation be as I launched a new website. This is really exciting and is definitively going to be one of my mid-carrier objective. I learned a lot about programming and especially the importance of architecture design. Thanks to Curse, my work was able to reach millions of players and this is probably the most satisfying part !