

Université de Toulouse

Master 1 GEP

« Géographie des changements Environnementaux et Paysagers » (**GEP**)

RAPPORT DE STAGE

Cartographie et modélisation graphique animée sur support web, test de développement appliqué à l'évolution du territoire de l'Observatoire Hommes-Milieus Pyrénées Haut Vicdessos (Ariège).



Abdillahi ADEN EGUEH

Maître de stage : Monsieur Laurent JEGOU, UMR LISST

Tutrice-enseignante : Emilie LERIGOLEUR, UMR GEODE

05/09/2017

Remerciements

Je tiens tout d'abord à remercier M. Laurent Jégou, mon maître de stage, qui est maître de Conférences au Département de Géographie, Aménagement et Environnement de l'Université de Toulouse Jean Jaurès, pour m'avoir accueilli et permis d'effectuer mon stage au sein du laboratoire LISST, ainsi que pour son encadrement tout au long de mon stage pour tous ses conseils et son expertise dans le domaine du Webmapping.

Je tiens à remercier particulièrement mon enseignante tutrice pour ses conseils, sa disponibilité, sa modestie et pour toute son aide ainsi que l'expérience et sa spécialité qu'elle m'a partagé.

Un grand merci également à l'ensemble de l'équipe de l'Atelier cartographie et de la salle informatique pour leur accueil et leur aide qui m'a été précieuse durant ce stage.

Enfin, j'ai le plaisir de remercier également ma famille qui m'a soutenu affectueusement durant ce long périple, ainsi que toutes les personnes qui m'ont conseillées, et qui ont partagé leurs connaissances et leur savoir-faire pendant ce stage.

SOMMAIRE

I) Introduction : mise en contexte et analyse de la commande

II) Présentation du travail technique, des problèmes et de leur résolution

III) Résultats et discussion

1 Introduction

Le webmapping ou la cartographie SIG en ligne, propose des solutions de SIG en ligne qui ont fait leur apparition il y a environ quelques décennies. Ce domaine est en pleine expansion même s'il est récent. Le webmapping est en essor grâce au développement des solutions open source.

Dans le cadre de la période de formation en milieu professionnel, j'ai effectué un stage de 8 semaines au sein du laboratoire LISST de l'université de Toulouse 2 Jean Jaurès. Durant mon stage j'ai pu compléter mes compétences en géomatique notamment le webmapping. Etant passionné par le webmapping, ce stage était pour moi, une passerelle pour découvrir le fonctionnement du milieu professionnel, tout en mettant en pratique les connaissances acquises en milieu universitaire.

L'objectif du stage est double, d'une part la production d'une carte interactive comportant l'évolution spatiale et temporelle du paysage sur le territoire de l'Observatoire Hommes-Milieus Pyrénées Haut Vicdessos. Cette carte doit "raconter" l'histoire de ce territoire à partir d'un tableau d'événements (annexe 1), objets spatiaux de type polygones, lignes ou points fourni par le commanditaire.

L'autre objectif est de tester techniquement s'il est possible de représenter ces événements par la combinaison d'éléments simples, pour en déduire un modèle graphique, correspondant à la méthode chorématique.

Ainsi, nous présenterons l'organisme d'accueil et l'objectif détaillé du stage ensuite nous détaillerons l'intégralité des méthodes utilisées et les solutions retenues. A la suite de la présentation des travaux, nous expliquerons les résultats et les discuterons.

1.1 Les laboratoires d'accueil : l'unité mixte de recherche Géographie de l'environnement (GEODE) et le Laboratoire Interdisciplinaire Solidarités, Sociétés, Territoires (LISST)

Durant mon stage j'étais amené à travailler dans deux laboratoires, le laboratoire GEODE et le laboratoire LISST qui sont des laboratoires spécialisés dans le domaine de la géographie de l'environnement et en sciences humaines et sociales.

Le laboratoire GEODE (Géographie de l'Environnement) est une unité mixte du CNRS (INEE) / Université de Toulouse 2 qui est composée de 37 membres permanents et de 30 Membres permanents de l'équipe et dont le directeur est Didier Galop. L'équipe est composée de :

Chercheurs CNRS :

- 2 directeurs de recherche, 7 chargés de recherche
- Enseignants-chercheurs : 4 professeurs, 1 professeur émérite, 12 maîtres de conférences (dont 1 HDR)
- ITA CNRS : 10
- Membres associés : 20 doctorants

Le laboratoire Géode travaille depuis sa création sur le rapport environnement/sociétés tout en mettant l'accent sur des concepts et méthodologies de la géographie et aussi de branches sociales, et historique. Ces chercheurs travaillent sur des disciplines de l'écologie, de géohistoire, de paléoécologie et d'archéologie ainsi que des experts en télédétection et modélisation spatiales ou multi-agents.

Quatre axes majeurs structurent l'activité scientifique :

Axe1: Anthropisation, paléoécologie et histoire de l'environnement

Axe2 : Dynamiques et enjeux contemporains de l'environnement et des paysages

Axe3 : Modélisation des trajectoires spatio-temporelles des paysages

Axe Transversal : Education au développement durable (EDD) Construction des savoirs scolaires en géographie

Grace à leurs travaux de recherches dans les domaines de la gestion durable, territoriale et de la prévention de risques, plusieurs membres du groupe font partie de comités nationaux d'expertises.

Quant au deuxième laboratoire où j'ai effectué mon stage, à savoir le laboratoire LISST (Laboratoire Interdisciplinaire Solidarités, Sociétés, Territoires), c'est un groupe Mixte de Recherche en Sciences Humaines et Sociales qui relève des sections 36, 38 et 39 du CNRS.

Ce laboratoire qui fait appel à plusieurs disciplines des Sciences Humaines et Sociales : anthropologie, études urbaines et rurales, géographie, sociologie également aussi économie, psychologie ou sciences de gestions. Ses activités s'inscrivent dans les équipes suivantes : le Centre d'Anthropologie Sociale, le Centre d'Études des Rationalités et des Savoirs, le Centre Interdisciplinaire d'Études Urbaines et Dynamiques Rurales, dans un dialogue transversal qui s'établit autour d'actions interdisciplinaires. L'unité met à leur service une plateforme d'accompagnement de ces recherches.

1.2 L'Observatoire Hommes-Milieux Pyrénées Haut Vicdessos et Haute vallée du Gave

Les Observatoires Hommes-Milieux (OHM) sont des dispositifs récents de recherche mis en place pour bien comprendre et étudier l'interaction Homme-milieu.

Les OHM se consacrent à l'étude de relation société-environnement mais également à l'étude des dynamiques de l'évolution socio-écologique dans des systèmes fortement anthropisés, Les Observatoires Hommes-Milieux sont des centres d'observation dans la mesure où l'une de leurs fonctions est d'avoir des réponses aux questions sociales, économiques, de santé, et politiques, a cet égard, ils peuvent constituer un dispositif d'aide à la décision politique (Par exemple l'étude d'une fermeture d'une exploitation minière ou d'un pont).

Les OHM sont organisés et soutenu par l'INEE, avec la collaboration des Zones Ateliers (ZA) et les Ecotrons.

Dans le monde, il en existe 11 OHM et dans le territoire national on en compte 6 dont 5 sont opérationnels :

- OHM Bassin minier de Provence
- OHM Pyrénées Haut-Vicdessos
- OHM Vallée du Rhône
- OHM Littoral Méditerranéen
- OHM – Oyapock (Guyane)

Dans notre cas, nous allons nous intéresser plus particulièrement à l'OHM Haut Vicdessos qui a été créée en 2009 sur un territoire montagnard situé en Ariège, marqué par un fort déclin des activités agro-pastorales, un exode rural conséquent et la fermeture des industries métallurgiques en 2003.

La vallée du Vicdessos représente un intérêt pour l'implémentation d'un Observatoire Homme-Milieu :

- des caractéristiques représentatives d'une grande partie des vallées montagnardes des Pyrénées mais aussi des originalités spécifiques, notamment pour la période contemporaine
- un acquis important en termes de recherche, en particulier dans tout le domaine de l'histoire de l'environnement, de la paléoécologie et de l'archéologie
- une proximité avec plusieurs centres universitaires et unités de recherches susceptibles d'intervenir dans les recherches (Foix, Toulouse, Montpellier, Pau)

Source : http://w3.ohmpyr.univ-tlse2.fr/presentation_ohm_pyr.php)

1.3 Objectifs du stage

1.3.1 Prototype de carte interactive pour les événements historiques sur le territoire du Haut Vicdessos

L'objectif général de ce stage se compose en deux grandes parties, le premier objectif principal est de réaliser un prototype qui permet de créer une carte interactive qui aiderait à expliquer les changements successifs des événements passés sur ce territoire, avec une vision diachronique.

Aussi, cela permettrait aux chercheurs du laboratoire GEODE de faciliter leurs recherches afin d'avoir une représentation plus claire de l'évolution spatiale des paysages en fonction du temps.

1.3.2 Chorèmes

Le second objectif de ce stage est d'une part de tester techniquement s'il est possible de développer des chorèmes simples et animés, et éventuellement de les combiner, à l'aide de la bibliothèque de fonctions Javascript D3.js, tout en respectant les règles générales de chorématique.

En outre, il faut également évaluer si cette méthode est accessible, avantageuse, pour les chercheurs qui travaillent sur le modèle chorématique. Pour cela nous avons choisi certains événements pour le test d'un chorème simple et animé puis de modèles plus complexes.

1.4 Démarche méthodologique globale

Cette méthodologie présente une approche globale et concrète des différentes étapes de la conduite du stage.

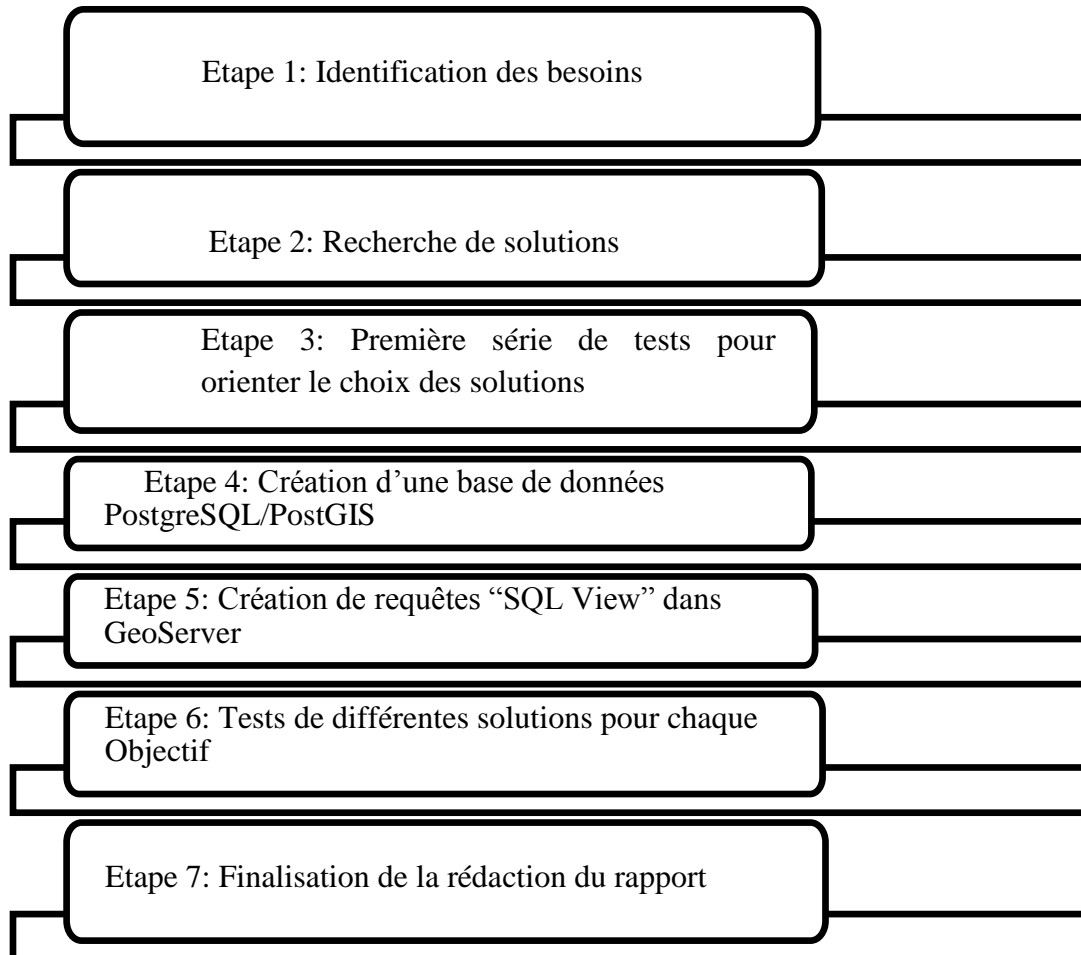
D'abord nous nous sommes organisés pour mener à terme les différentes phases du projet, en informant périodiquement mon maître de stage et ma tutrice-enseignante de l'état d'avancement du projet, afin d'échanger sur les difficultés rencontrées lors de l'exécution des différentes tâches à réaliser. Ceci m'a permis d'avoir personnellement une vision d'ensemble de la méthodologie et du déroulement du travail.

Les trois grandes phases du projet sont les suivantes :

- ✓ mise en contexte et analyse de la commande
- ✓ Présentation du travail technique, des problèmes et de leur résolution
- ✓ Résultats et discussion

Ensuite afin d'assurer une organisation rationnelle et cohérente du projet, j'ai adopté un plan de management divisé en étapes successives avec l'accord de mon maitre de stage et mon enseignante tutrice. Au cours de chacune de ses étapes plusieurs vérifications ont été réalisées par nous pour

assurer de la conformité avec la fiche méthodologique du projet. Donc nous avons établi un organigramme de méthodologique:



2 Présentation du travail technique, des problèmes et de leur résolution

Nous avons présenté dans la partie précédente, le contexte général du stage et les objectifs (prototype de carte interactive diachronique et modèle graphique chorématique). C'est dans ce cadre que nous allons aborder les différentes méthodes que nous avons utilisées pour élaborer le prototype. Ensuite nous expliquerons les procédures de la réalisation du modèle chorématique, en commençant par le chorème simple jusqu'au chorème complexe.

2.1 Objectif “prototype de carte interactive pour l’OHM

2.1.1 Identification des besoins

Dans la plupart des expériences courantes et ceci, quelque soit le domaine, la recherche de solutions au problème ou l'identification des besoins nécessite très souvent de répondre aux mêmes questions en général : on utilise classiquement la méthode (QQOQCCP) ou Loi de Quintilien qui permet d'y répondre.

Cette méthode a ainsi été utilisée (figure 1, page suivante) lors des entrevues avec Émilie LERIGOLEUR. Cette méthode nous a permis d'avoir une vision générale de notre prototype ainsi que les outils qui pourraient nous être utiles. À ce moment là, nous avons classé les fonctionnalités attendues du prototype par priorité pour avoir une idée plus claire de l'architecture de ce prototype. Nous avons retenu la liste des fonctionnalités suivantes :

- afficher des couches vectorielles et sur un ou plusieurs fond(s) de carte.
- Proposer un curseur temporel avec des étapes qui sont liées à des événements marquants (non réguliers dans le temps).
- Offrir la possibilité d'ajouter de nouveaux objets spatiaux (événements).
- Offrir la possibilité de zoomer et dézoomer les objets spatiaux.
- Prévoir l'apparition d'infobulles lorsqu'on clique les objets spatiaux.

Quoi	carte interactive	
Qui	l'administrateur: Abdillahi ADEN EGUEH après sera Emilie LERIGOLEUR Les utilisateurs: les membres de l'OHM	sécuriser l'accès
Où	serveur du laboratoire GEODE	
Quand	13 juillet la carte doit être opérationnelle	
Comment	moyens matériels : mon pc portable avec serveur web et base de donnée installés (transfert futur vers serveur) Méthode: Programmation web et webmapping	
Combien	Aucun budget	
Pourquoi	pour que les chercheurs de membres de l'OHM puissent consulter la carte et éventuellement ajouter de nouveaux événements	évolution vers une carte participative ?

Figure 1:Utilisation de la Loi du Quintilien pour affiner les besoins

2.1.2 Recherche de solutions

Naturellement, il existe plusieurs façons de concevoir un prototype mais il fallait qu'on cherche des solutions qui puissent répondre aux fonctionnalités fixées (section 2.1.1), pour cela nous avons choisi les solutions suivantes :

- Concernant le serveur de cartographie nous avons comparé plusieurs serveurs afin de déterminer celui qui correspond le mieux à notre besoin. Notre choix s'est porté sur l'application GeoServer car elle prend en charge l'attribut TIME sur webservice WMS mais propose également l'utilisation de vues (SQL View) qui permettent d'exécuter une requête SQL personnalisée à chaque demande de carte WMS, sur une même couche sans créer de table dans la base de données.
- Le choix du client cartographique est par ailleurs très important pour la représentation entre l'utilisateur et le serveur. Il existe plusieurs applications clientes, dont deux principales, étudiées afin de sélectionner la plus adaptée à notre besoin.

- Leaflet est une bibliothèque de fonctions JavaScript libre de cartographie en ligne. Relativement à notre premier test, nous avons choisi Leaflet, mais nous avons constaté que le plugin *time dimension* ne nous donne pas le curseur temporel.
- L'autre bibliothèque de fonctions JavaScript testée est OpenLayers. Elle possède une grande popularité et propose de nombreuses fonctionnalités. Nous avons choisi finalement OpenLayers, car elle nous a permis de créer le curseur temporel sans passer par un plugin, grâce au code JavaScript, à l'html 5 et au Css.

2.1.3 Première série de tests pour orienter le choix des solutions

Le but du premier test était de savoir si GeoServer prend en charge un attribut « temps » dans les requêtes WMS GetMap (figure 3). Pour cela nous avons correctement configuré un jeu de test avec une dimension temporelle (figure 2). Ce dernier sert à spécifier un sous-ensemble temporel pour requêter en fonction du temps. On configure les attributs de temps dans GeoServer à partir de l'interface Web GeoServer en naviguant dans l'onglet Couches -> Dimensions (figure 2).

À partir de ce test, nous avons déduis que GeoServer prend en charge les attributs de temps dans les requêtes GetMap et cela fonctionne correctement (figure 4). À ce moment là, nous avons essayé d'ajouter un curseur temporel. Pour Leaflet, nous avons trouvé un plugin qui proposait un curseur temporel mais, malheureusement, on n'a pas pu adapter le code source selon nos besoins. Donc nous avons remplacé Leaflet par OpenLayers avec lequel nous avons réussi à créer un curseur temporel sans plugin mais avec du code JS, HTML et CSS. C'est ainsi que notre choix définitif s'est porté sur OpenLayers pour la création de notre prototype.

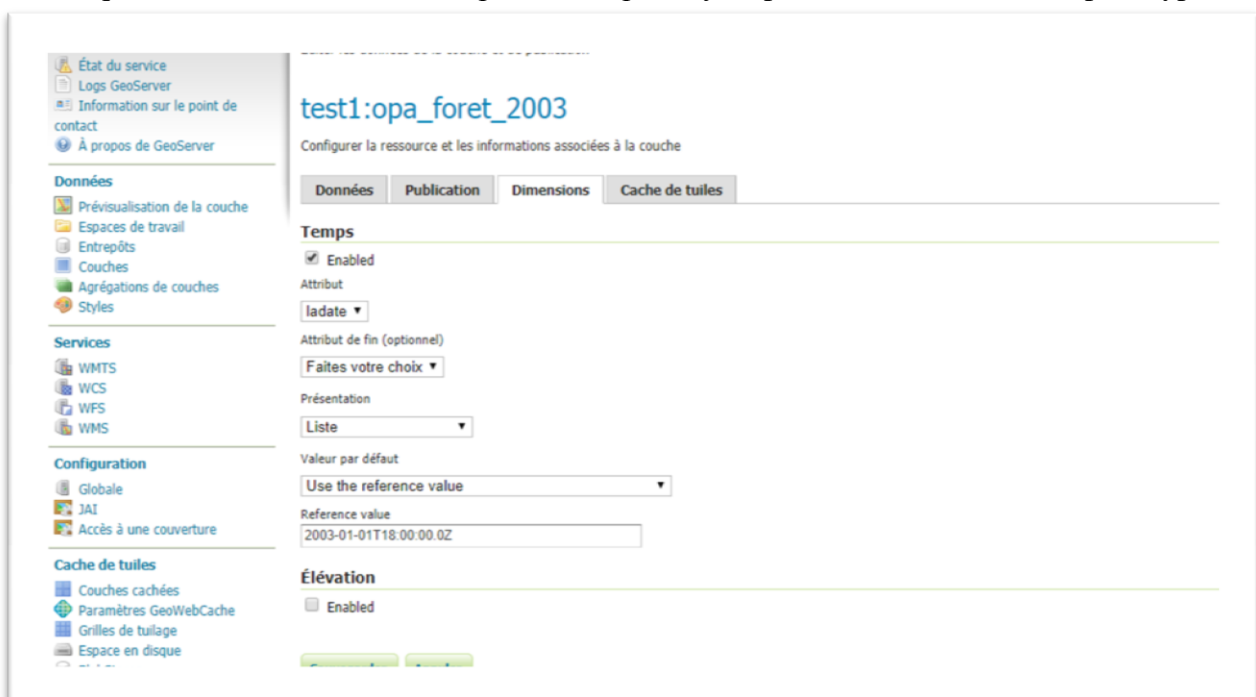


Figure 2: Configuration de dimension TIME pour une couche WMS

```
<script type="text/javascript">
function init(){
  var map = L.map('map').setView([42.77, 1.49], 12);
  var nexrad = L.tileLayer.wms("http://localhost:8080/geoserver/wms", {
    layers: 'opa_foret_2003,opa_foret_2008',
    format: 'image/png',
    transparent: true,
    time: '2002-12-31T00:00:00.0Z/2009-01-02T00:00:00.0Z',
    attribution: "Weather data © 2012 IEM Nexrad"
  });
  map.addLayer(nexrad);
}
```

Figure 3: Script du test 1 avec Leaflet

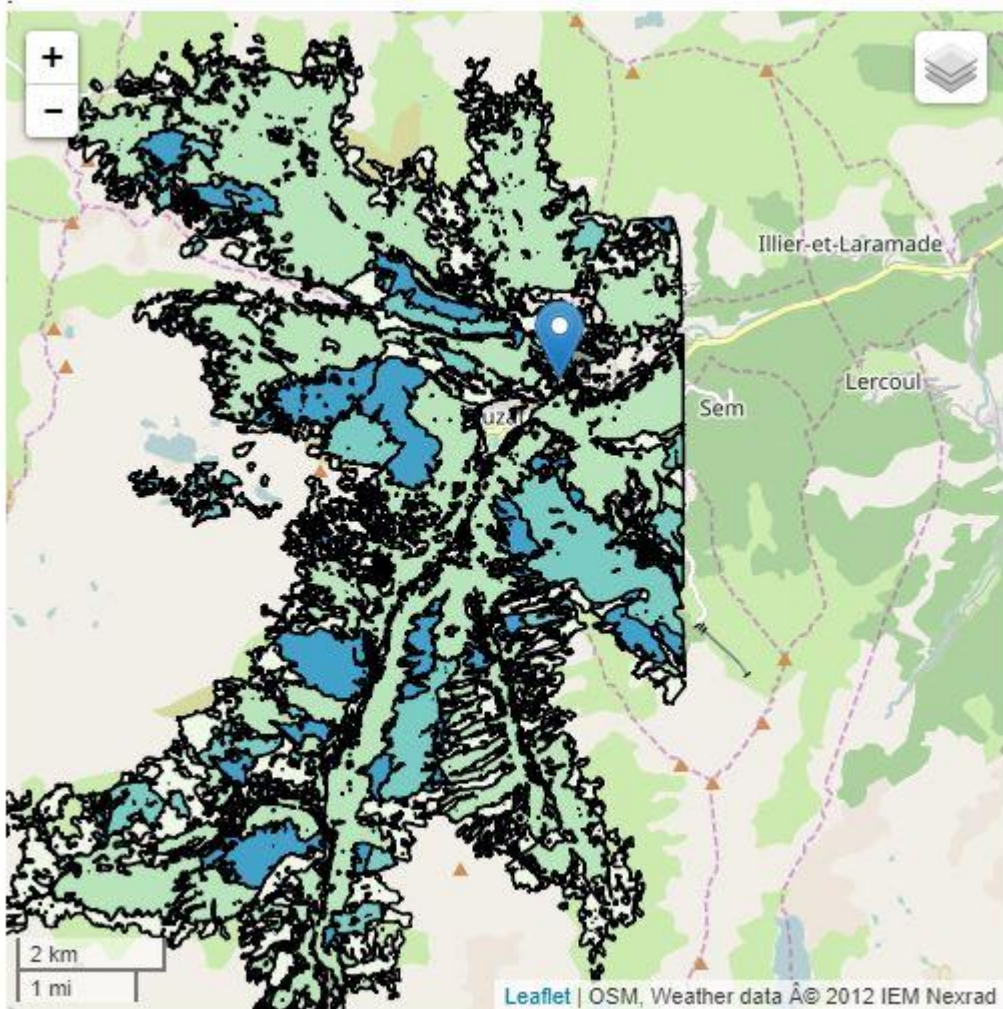


Figure 4: Résultat du premier test avec Leaflet

2.1.4 Installation des logiciels GeoServer et PostgreSQL

Pour la réalisation de ce prototype, cela a nécessité d'installer certains logiciels comme le moteur cartographique GeoServer et le serveur de bases de données PostgreSQL qui possèdent quelques spécificités lors de leur installation.

- ❖ GeoServer est un logiciel open source téléchargeable gratuitement mais avant l'installation il faut bien vérifier qu'un Java Runtime Environment (JRE) soit installé sur le système. GeoServer nécessite un java 8 qui est téléchargeable sur le lien suivant <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Lors de l'installation de GeoServer il faut savoir trois points importants :

- L'interface d'administration Web de GeoServer nécessite une authentification pour la gestion de l'interface. Donc il faut bien définir le nom de l'utilisateur et le mot de passe et bien le conserver. Les paramètres par défaut sont admin / GeoServer (figure 5). Il est important de les modifier à partir des valeurs par défaut pour avoir plus de sécurité.
- Deuxième point essentiel, noter le port http choisi, nous avons utilisé celui qui est par défaut « 8080 » (figure 6) mais il est possible de changer le port tant que ce port n'est pas utilisé par un autre logiciel serveur.
- Enfin, il faut choisir si GeoServer doit être exécuté manuellement ou installé en tant que service (figure 7). Pour ce prototype nous avons sélectionné d'exécuté manuellement pour optimiser la performance de l'ordinateur.



Figure 5: définition du nom de l'utilisateur et son mode de passe pour l'administrateur

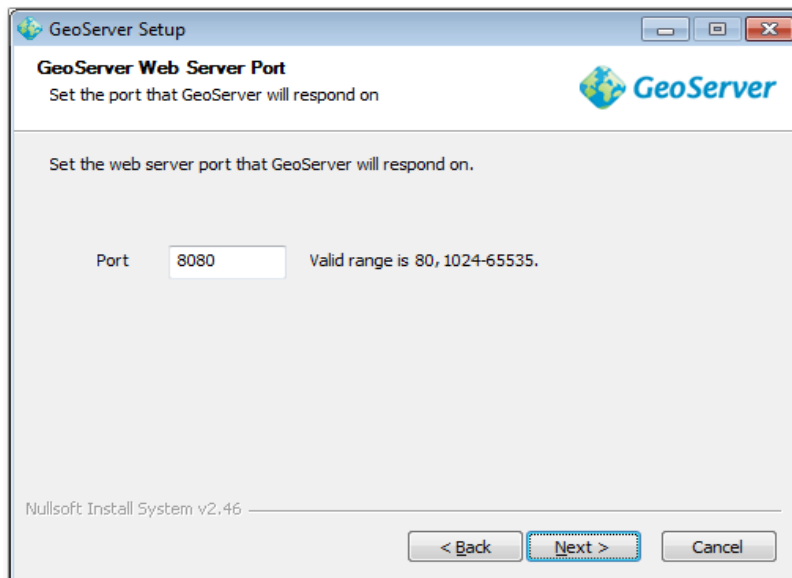


Figure 6: Définition de port de GeoServer

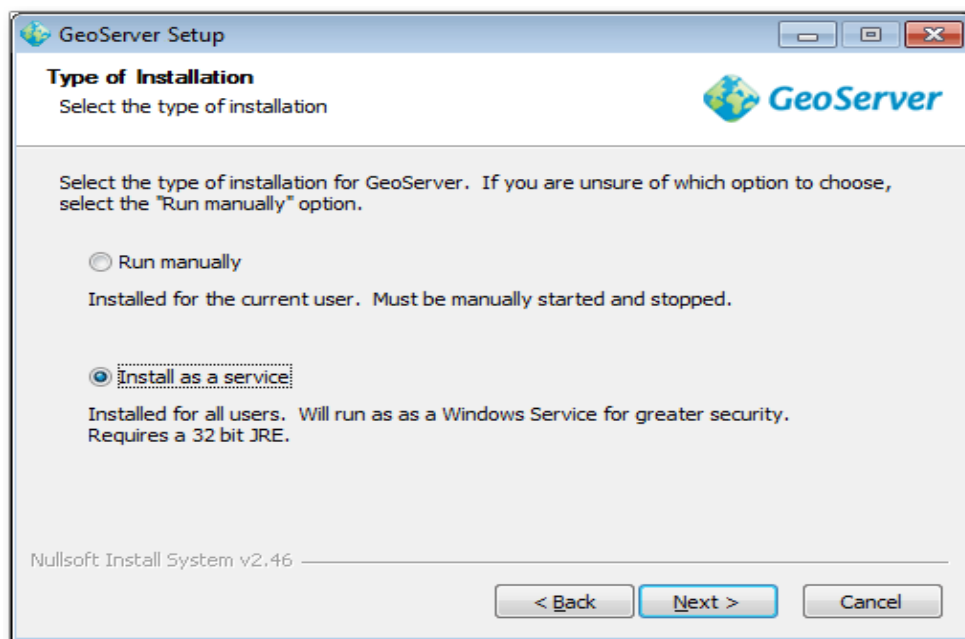


Figure 7:: Installation de GeoServer en tant que service

- ❖ L'installation du logiciel PostgreSQL est un peu délicate mais j'ai suivi le tutoriel d'installation des cours de madame Émilie Lerigoleur, selon trois étapes :
 - Pour l'installation des binaires, il faut bien choisir la version de PostGIS ainsi que la version de votre système d'exploitation (figure 8).
 - Plusieurs outils sont installés en même temps que PostgreSQL, notamment les deux interfaces d'utilisation que nous allons utiliser : L'interface graphique pgAdminIII et le shell SQL en ligne de commande (figure 9). Ces deux outils

permettent de créer, gérer et administrer PostgreSQL et ses bases de données spatiales ou non spatiales. Pour créer la première base de données, il suffit de copier le code (figure 10).

- Afin d'activer l'extension il faut au préalable créer une base de données dans PostgreSQL via la commande « CREATE DATABASE ». Ensuite, il suffit de copier et d'exécuter la requête (figure 10) « CREATE EXTENSION PostGIS » pour activer PostGIS et transformer cette BDD en base spatiale.

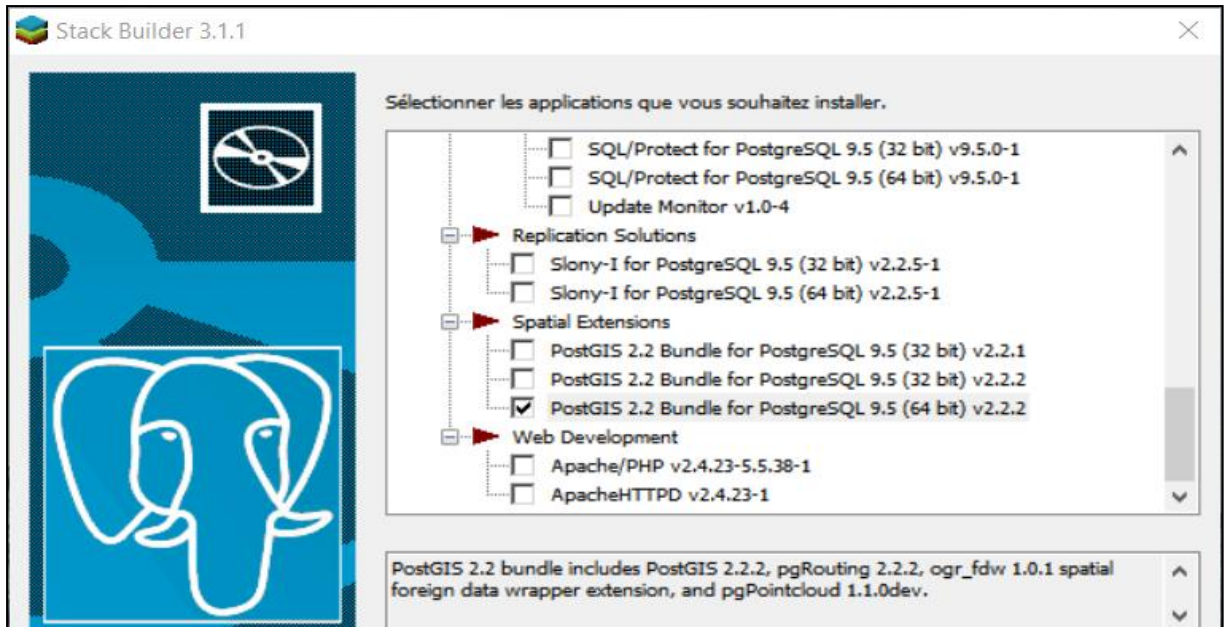


Figure 8: installation des binaires

```
C:\Program Files\PostgreSQL\9.5\bin\psql.exe
psql (9.5.4)
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
Les caractères 8 bits peuvent ne pas fonctionner correctement.
Voir la section « Notes aux utilisateurs de Windows » de la page
référence de psql pour les détails.
Saisissez « help » pour l'aide.

postgres=# CREATE DATABASE bdd_postgis;
CREATE DATABASE
postgres=# \connect bdd_postgis
Attention : l'encodage console (850) diffère de l'encodage Windows (1252).
Les caractères 8 bits peuvent ne pas fonctionner correctement.
Voir la section « Notes aux utilisateurs de Windows » de la page
référence de psql pour les détails.
Vous êtes maintenant connecté à la base de données « bdd_postgis » en tant qu'utilisateur « postgres ».
bdd_postgis=# CREATE EXTENSION postgis;
CREATE EXTENSION
bdd_postgis=#
```

Figure 9: requête d'activation de l'extension PostGIS

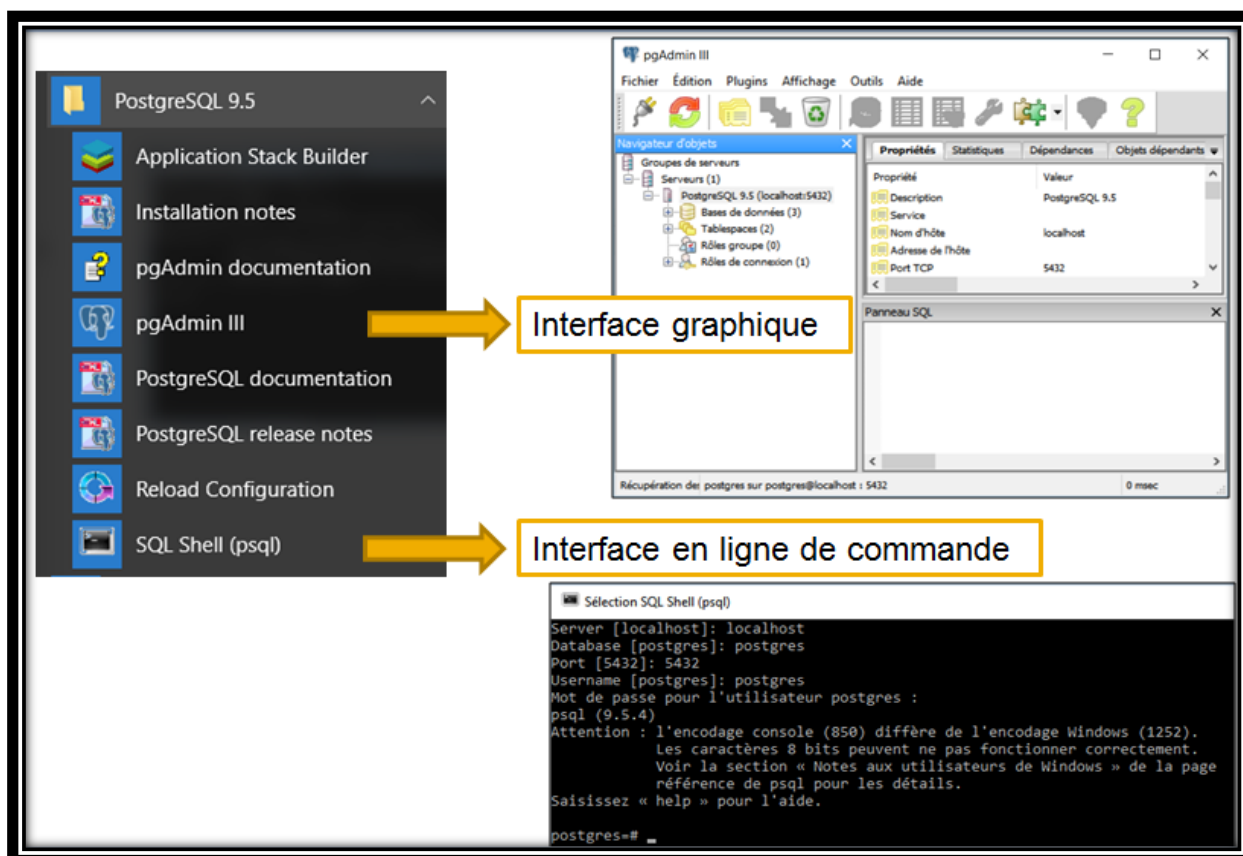


Figure 10: Interface de PgAdminIII

2.1.5 Création d'une base de données PostgreSQL/PostGIS

Suite au premier test d'orientation du choix des outils, nous avons décidé de créer une base de données spatiale. Le but est d'une part d'optimiser GeoServer lors de la requête GetMap, c'est-à-dire d'accélérer l'affichage des cartes pour le côté client. D'autre part, cette base permettra d'exploiter la puissance du SQL pour interroger les données selon nos besoins, afin de pouvoir effectuer la requête dans un script PHP, que nous détaillerons dans la section 3.2.2.

Premièrement, nous avons créé notre base de données à partir de l'interface graphique pgAdminIII (figure 11). Cependant, il existe plusieurs façons d'importer les fichiers shapefile, nous avons choisi d'importer nos couches à partir de Qgis grâce au plugin DB manager (figure 12).

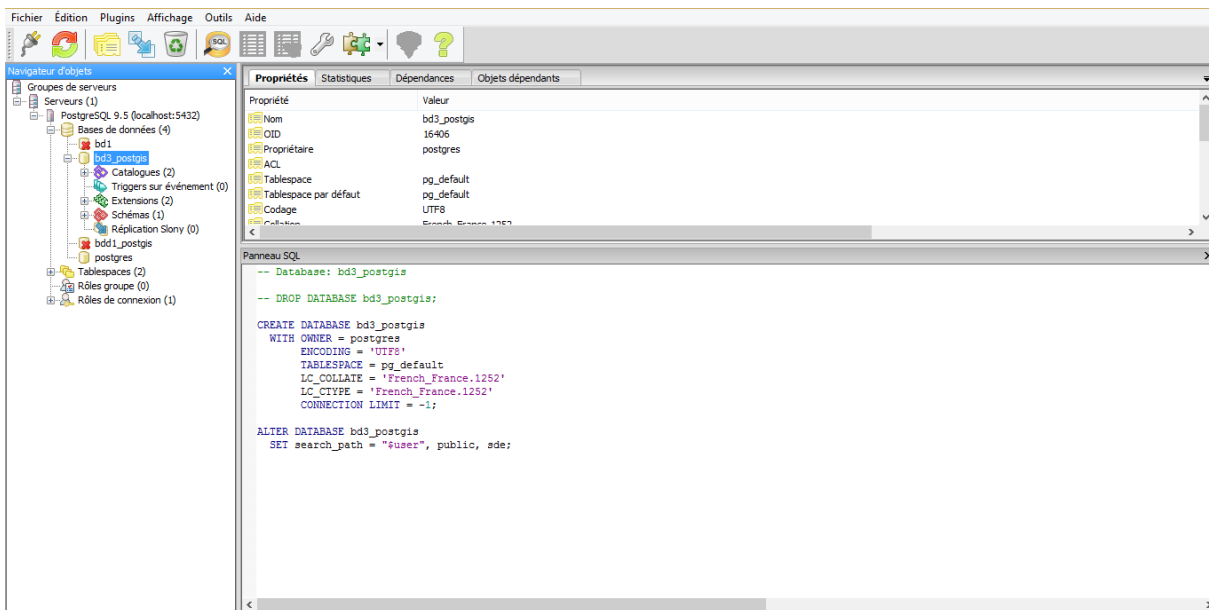


Figure 11: création de base de donnée « bdd3_postgis » sur l'interface graphique de pgAdminIII

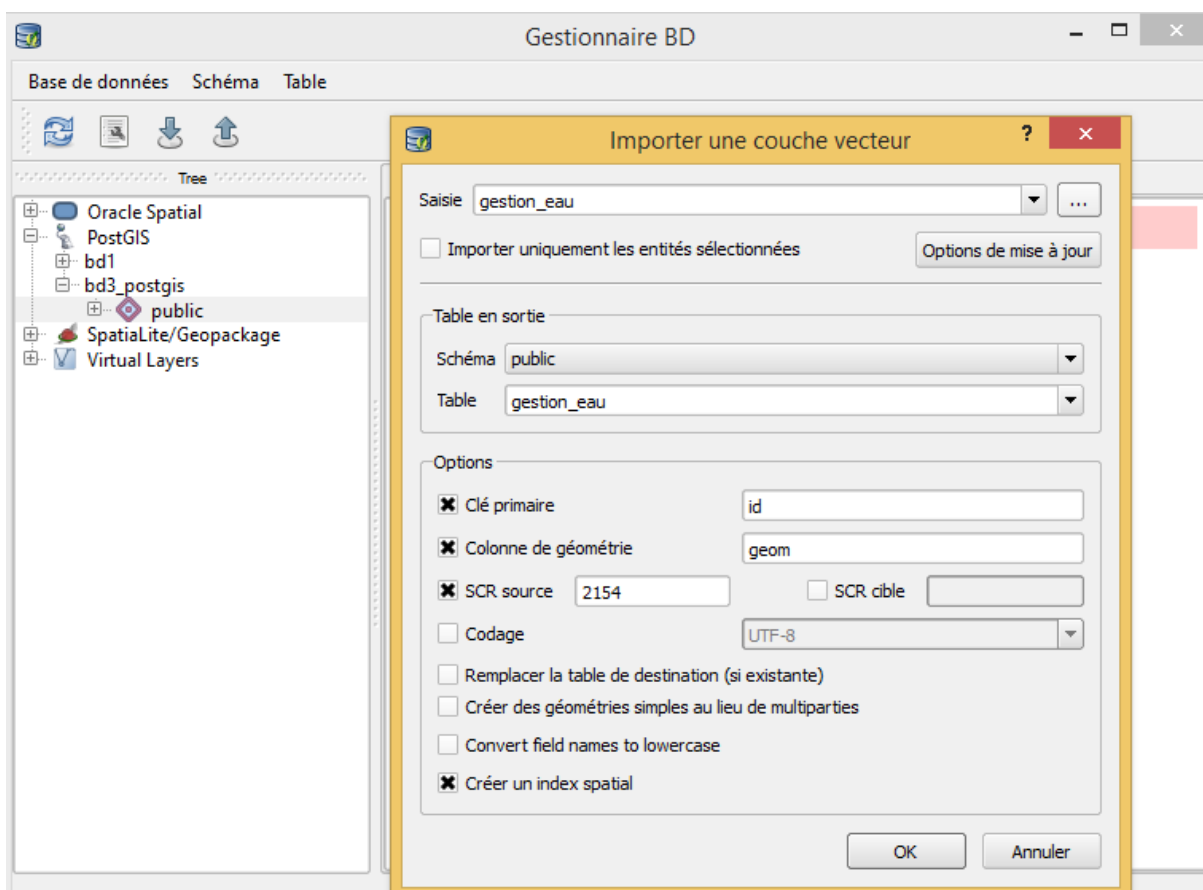


Figure 12: Importation d'une couche shapefile

2.1.6 Création de requêtes “SQL View” dans GeoServer

Après avoir vérifié que GeoServer prenne bien en charge l’attribut **TIME**, nous avons également pu valider une autre solution, l’utilisation de **SQL View dynamique**. Cette méthode permet de définir les couches comme **des vues SQL** à partir de GeoServer 2.1.0. On peut ainsi exécuter une requête SQL personnalisée et paramétrée à chaque demande sur une même couche sans préalablement créer une vue dans la base de données. Cela est un grand avantage pour nous car elle optimise le temps de chargement des couches depuis GeoServer et permet de le paramétrer.

En effet, les requêtes SQL View peuvent être personnalisées par des paramètres spécifiques en les configurant à partir de l’interface web de GeoServer. Les valeurs de ces paramètres peuvent être fournies d’une manière dynamique dans la requête GETMap du protocole WMS. Comme nous traitons des événements non réguliers dans le temps, on s’intéresse seulement à **l’année** donc nous avons décidé de classer les événements en deux catégories :

- ❖ **Les événements *continus*** sont des événements qui durent plus d’un an dans notre cas. Ainsi ces événements possèdent une date début et une date de fin, donc nous avons cherché une requête SQL View qui permette d’afficher les périodes voulues, lorsque l’utilisateur clique sur une année qui se trouve dans cet intervalle. Le but de cette requête permet d’abord, d’extraire les années de **date de début** et de **date de fin**, ensuite elle vérifie avec le paramètre « **%annee %** » si le nombre est compris dans cet intervalle, sinon elle affiche l’inverse (figure 13).

Nom de la vue
usine3

Directive SQL

```
select * from usine where %annee% between
extract(YEAR FROM date_deb) And extract(YEAR FROM
date_fin)
```

Paramètres de la vue SQL
Deviner les paramètres à partir du SQL Ajouter un nouveau paramètre Retirer la sélection

<input type="checkbox"/> Nom	Valeur par défaut	Expression régulière de validation
<input type="checkbox"/> annee	1908	^[w\d\s]+\$

Échapper les caractères spéciaux SQL

Attributs
Actualiser Deviner le type de géométrie et le système de référence spatial

Nom	Type	SRID	Identifiant
id	Integer		<input checked="" type="checkbox"/>
geom	MultiPolygon	4326	<input type="checkbox"/>
name	String		<input type="checkbox"/>
descriptio	String		<input type="checkbox"/>

Figure 13: Configuration d’une requête SQL View d’un événement ponctuelle

- ❖ **Les événements *ponctuels*** sont des événements qui durent moins d'un an pour ce prototype. Pour ceux-ci nous avons spécifié une requête SQL View simple qui permet de vérifier si les **dates de début** sont égales au paramètre « **%annee%** » pour afficher les résultats (figure 14).

Nom de la vue

Directive SQL

```
select * from ev_hydro3 where extract(YEAR FROM annee)=%annee%
```

Paramètres de la vue SQL
 Deviner les paramètres à partir du SQL Ajouter un nouveau paramètre Retirer la sélection

<input type="checkbox"/> Nom	Valeur par défaut	Expression régulière de validation
<input type="checkbox"/> annee	<input type="text" value="1725"/>	<input type="text" value="^[w\dls]+\\$"/>

Échapper les caractères spéciaux SQL

Attributs
 Actualiser Deviner le type de géométrie et le système de référence spatial

Nom	Type	SRID	Identifiant
id	Integer		<input checked="" type="checkbox"/>
geom	<input type="text" value="MultiPolygon"/>	<input type="text" value="4326"/>	<input type="checkbox"/>
nom	String		<input type="checkbox"/>
annee	Date		<input type="checkbox"/>
date_fin	Date		<input type="checkbox"/>

Figure 14: Configuration d'une requête SQL View d'un événement continue

2.1.7 Tests de différentes solutions

2.1.7.1 Test 1 : Gestion des styles d'affichage

Il existait deux possibilités pour gérer le style des couches : soit, en programmant au niveau du client c'est-à-dire en JavaScript, soit utiliser un fichier SLD qu'on a créé à partir de Qgis dans GeoServer. On a choisi la deuxième option pour gagner plus de temps lors d'affichage des requêtes WMS.

D'abord, nous avons préparé le style de chaque couche dans Qgis puis nous avons exporté les styles assemblés comme un fichier SLD. Ensuite, nous avons configuré le fichier SLD dans l'onglet Style de GeoServer afin de choisir l'espace de travail et de vérifier s'il ya une erreur de syntaxe dans le code source (figure 15).

Enfin nous avons affecté à chaque couche son style correspondant, dans l'onglet « publication » (figure 16).

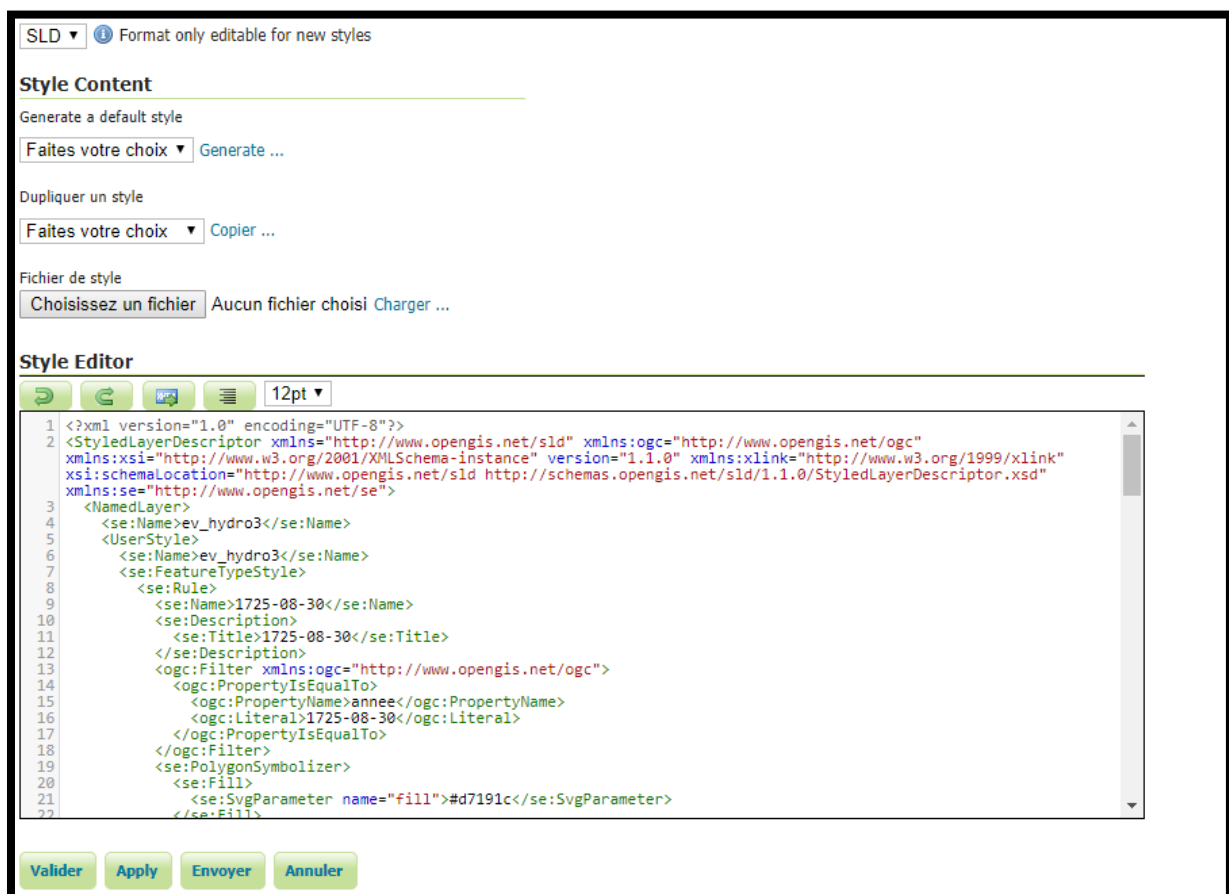


Figure 15: Configuration de style SLD dans le GeoServer

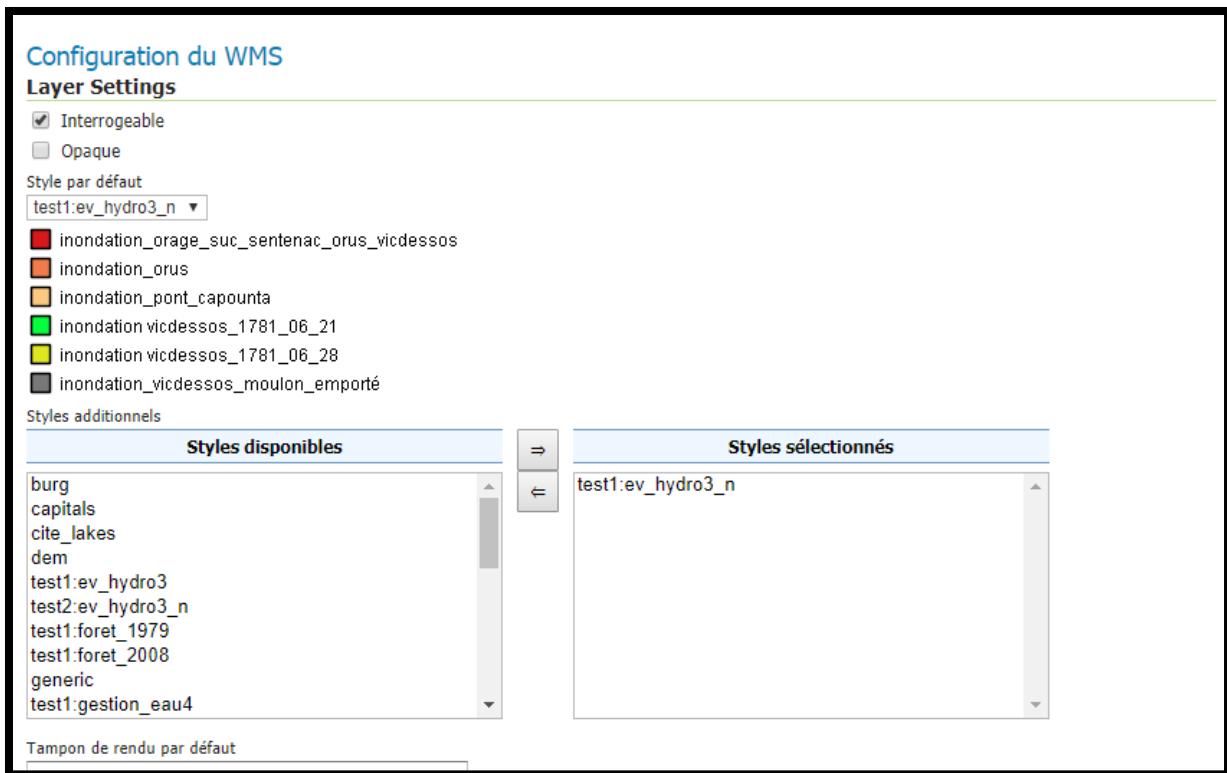


Figure 16: Affectation des styles dans GeoServer

2.1.7.2 Test2 : ajout d'un curseur temporel

Nous avons créé un curseur temporel à partir des codes JS, html 5 et CSS car c'est la méthode la plus simple et la plus rapide pour créer un curseur temporel simple (figure 17). Cette méthode n'a pas la possibilité de récupérer directement les années disponibles dans la base de données, ce qui nous a obligés de créer un objet « datalist » pour remplir manuellement les années disponibles dans la base de données (figure 17). Par ailleurs, malheureusement ce curseur est compatible seulement pour navigateurs chrome pour l'instant, dans Mozilla, le contenu est affiché d'une manière illisible (figure 18).

Pour contourner ce problème, nous avons utilisé le plugin **bootstrap-slider** pour créer un curseur. Ce curseur est compatible avec tous les navigateurs et on le voit correctement (figures 19 et 20). En outre, il a un grand avantage pour nous car il récupère directement les années disponibles dans la base de donnée grâce à une requête Php.

```
<datalist id="tickmarks">
  <option value="1725">
  <option value="1753">
  <option value="1772">
  <option value="1776">
  <option value="1781">
  <option value="1908">
  <option value="1914">
  <option value="1940">
  <option value="1980">
```

```
<option value="1984">
<option value="2003">
<option value="2006">
<option value="2009">
<option value="2017">

</datalist>
<input type="range" id="range" value="1725" max="2017" min="1725" list="tickmarks"
onChange="update_date()"/>&nbsp;&nbsp;&nbsp;<span id="libelle">1725</span></br></br>
```

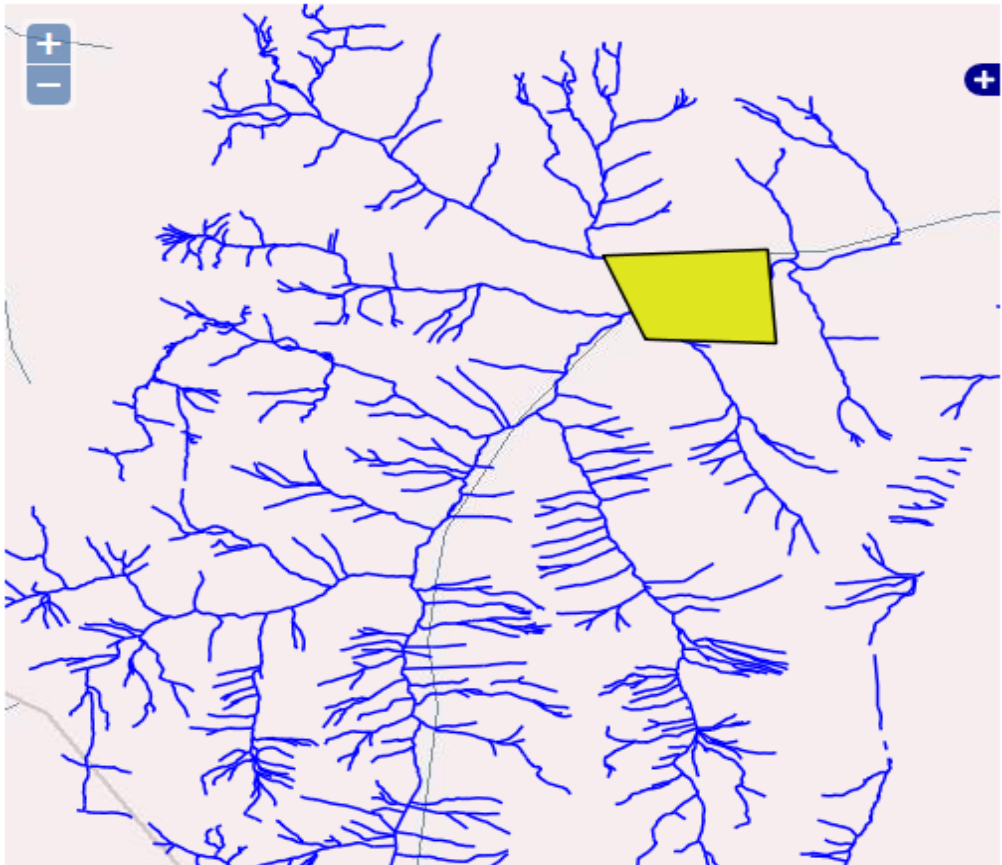
Figure 17:code source du curseur html

WMS Time Example

wmst, wms-t

Shows the use of the layer WMS-T (time) layer

1781



The screenshot displays a web-based interface for a WMS Time Example. At the top, the text 'WMS Time Example' is prominently displayed. Below it, the layer identifiers 'wmst, wms-t' are shown. A descriptive sentence reads 'Shows the use of the layer WMS-T (time) layer'. A horizontal time slider is positioned below the text, with a vertical arrow indicating the current time value, which is '1781'. The main part of the interface is a map showing a complex network of blue lines representing a river system. A specific area of the river network is highlighted with a yellow rectangular box. On the left side of the map, there are zoom controls consisting of a '+' button above a '-' button. On the right side, there is another '+' button. The entire interface is enclosed in a black border.

Figure 18:curseur illisible dans le navigateur Mozilla

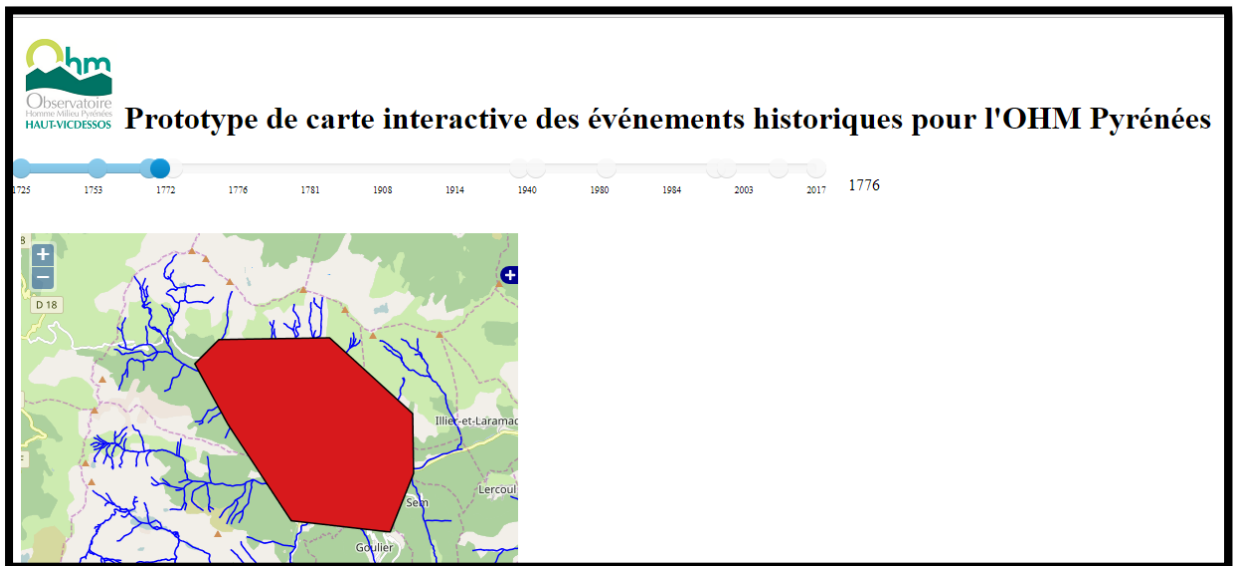


Figure 19: curseur compatible avec le navigateur chrome

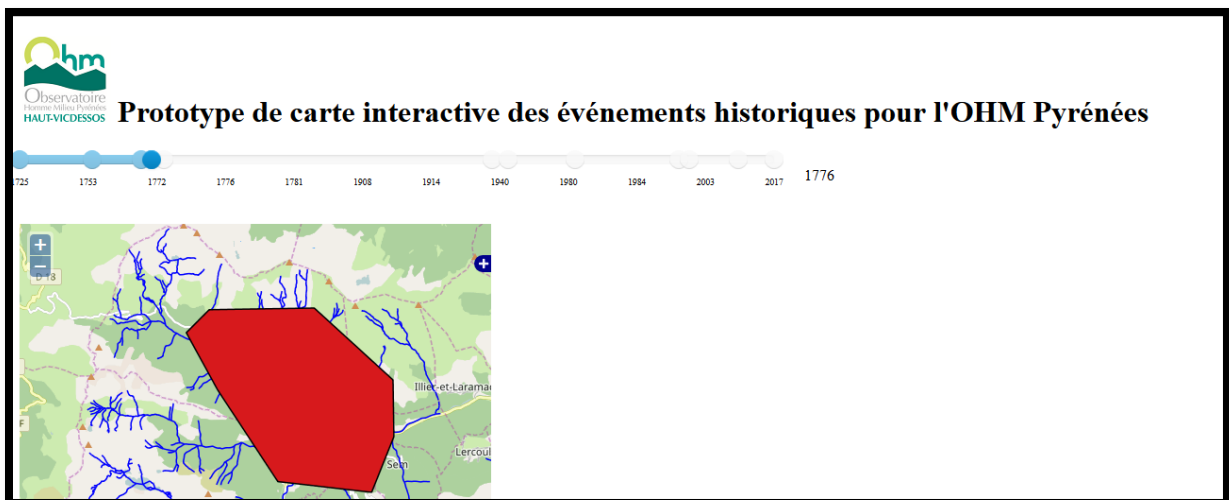


Figure 20: Curseur compatible avec le navigateur Mozilla

2.2 Objectif “chorèmes”

2.2.1 Identification des besoins

La chorématique est une méthode de modélisation géographique qui a été fondée par Roger Brunet en 1980 (R. Brunet, la carte de modèles et les chorèmes, 1984). Cette méthode permet de créer des modèles graphiques qui représentent un espace et leurs phénomènes spatiaux.

En général, il existe une table de 28 chorèmes considérées comme les règles de base des chorèmes, cette table a été créée par Roger Brunet (figure 21).

Pour identifier les besoins, nous avons hiérarchisé les événements en plusieurs catégories afin de schématiser ceux qui sont représentables dynamiquement (figure 22). Donc nous avons choisi de schématiser les événements suivants :

- Nouvelle Communauté de communes de la Haute-Ariège en 2017.
- Écobuage illégal dans le bassin versant de Bernadouze.
- Premières usines d'aluminium à Auzat.
- Fermeture de l'usine d'aluminium Péchiney.
- Inondation : le Vicdessos emporte le pont de Capounta à Auzat.

	POINT	LIGNE	AIRE	RÉSEAU
maillage				
	chef-lieu	limite administrative	État, région...	centres, limites et polygones
quadrillage				
	tête de réseau carrefour	voies de communication	aire de desserte irrigation, drainage	réseau
attraction				
	points attirés satellites	lignes d'isotropie orbites	aire d'attraction	liaisons préférentielles
contact				
	point de passage	rupture, interface	aires en contact	base tête de pont
tropisme				
	flux directionnel	ligne de partage	surfaces de tendance	dissymétries
dynamique territoriale				
	évolutions ponctuelles	axes de propagation	aires d'extension	tissu du changement
hiérarchie				
	semis urbain	relation de dépendance limites administratives	sous-ensemble	réseau maillé

Source : R. Brunet, Mappemonde 86/4, 1986

Figure 21 : la table des 28 chorème de base. Source (R.Brunet, 1986)

2.2.2 Recherche de solution

L'objectif principal n'est pas de chercher une solution définitive et complète, mais plutôt de tester si on peut représenter des chorèmes animés, avec l'aide de la bibliothèque de fonctions d3.js. Donc nous utiliserons cette bibliothèque durant la réalisation du projet. Il faut également que la schématisation du modèle soit la plus proche possible de la réalité tout en respectant les règles générales de la méthode et en suivant les 28 chorèmes de base. Nous avons réfléchi à la manière de représenter ces événements dont on a récapitulé les différents cas possibles dans le tableau ci-dessous (figure 22).

Cas possibles (id_evt)	Catégorie d'événements	Type de représentation	Evolution d'un objet spatial	Remarque
1, 16, 28, 30, 38	Apparition disparition des activités métallurgiques puis apparition des activités des loisirs	polygone	Taille fixe apparition et transformation (changement des couleurs)	Événement important
2, 3, 4, 13	des flux aux frontières	Flèche ??	Une flèche avec animation	
5, 7, 8, 9, 10, 11	Inondations	des lignes avec débordement de l'eau	Agrandissement d'épaisseurs puis retour à la normale	Se caler avec le cours d'eau ??
27, 33	Dépollution	Point	Info-bulle (apparition et disparition)	Point par-dessus par des autres événement qui sont un polygone
19	Tempête de neige	Point	Info-bulle (apparition et disparition)	
6, 12, 14, 15	Événements divers qui impactent tous la vallée	Polygone	Info-bulle ??	
17, 18, 20,22	Mise en service des centrales électriques	Point	Info-bulle (apparition et disparition)	
21, 24, 25	Mise en place du lac-barrage	Polygone	Info-bulle (apparition et disparition)	
26, 29, 31, 34	Cyclisme	Point	Info-bulle (apparition et disparition)	
35	Coupe forestière	Polygone	Changement de texture	
37	Feu	Polygone	Un polygone qui se propage	
36	Changement de limite de Communauté des communes	Polygone	Changement de forme	Changement d'échelle spatiale

Figure 22: Style de représentation des événements

2.2.3 Tests de différentes solutions

2.2.3.1 Test 1 : représentation des chorème simple

Comme nous avons déterminé de quelle manière certains événements sont représentables, alors nous avons pu commencer à schématiser l'événement le plus simple, celui de l'usine. Les premières usines d'aluminium à Auzat créées en 1908 ont fermé en 2003, mais l'endroit a changé de fonction et donc de nom. Donc nous avons décidé de schématiser un cercle dans lequel on va faire une transition tout en changeant légèrement la couleur du bleu en rouge (figure 23).

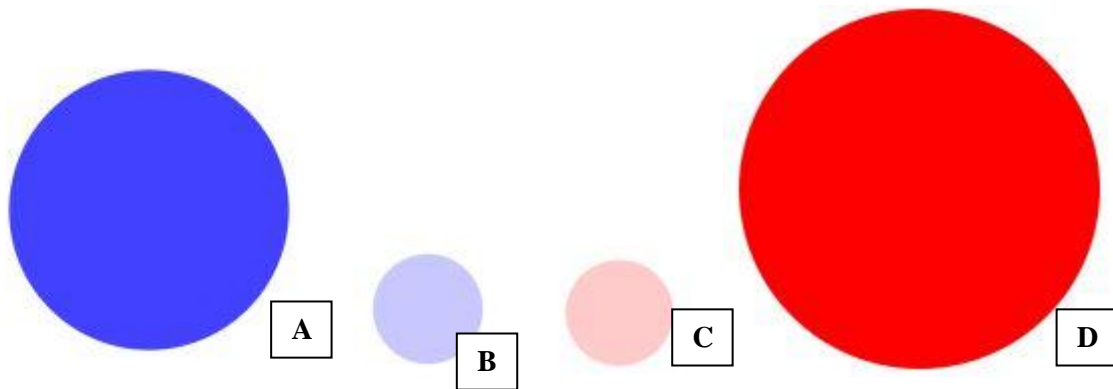


Figure 23: Modélisation de l'événement « ouverture et fermeture de l'usine »

2.1.1.1 Test 2 : schématisation d'un model chorématique animé

Nous continuons à tester mais cette fois-ci avec un modèle chorématique c'est-à-dire un modèle qui contient plusieurs événements, en essayant de représenter une situation complexe par la composition d'éléments simples, schématisés et animés.

Comme nous avons déjà défini les styles de présentation des événements dans le tableau précédent (figure 22), nous avons schématisé cette fois l'événement de l'inondation en essayant de faire l'agrandissement des épaisseurs puis un retour à la normale (figure 24), le cercle jaune représentant la propagation des feux et le cercle vert symbolisant la forêt (figure 24). Pour les incendies forestiers nous avons modélisé un cercle jaune qui se déplace vers le cercle vert et cela traduit la propagation des feux à la forêt (figure 22). Cependant, on s'est rendu compte que cette modélisation ne correspondait pas réellement la propagation du feu. Ainsi nous avons trouvé une autre solution, que nous détaillerons sur la section 3.3.1.

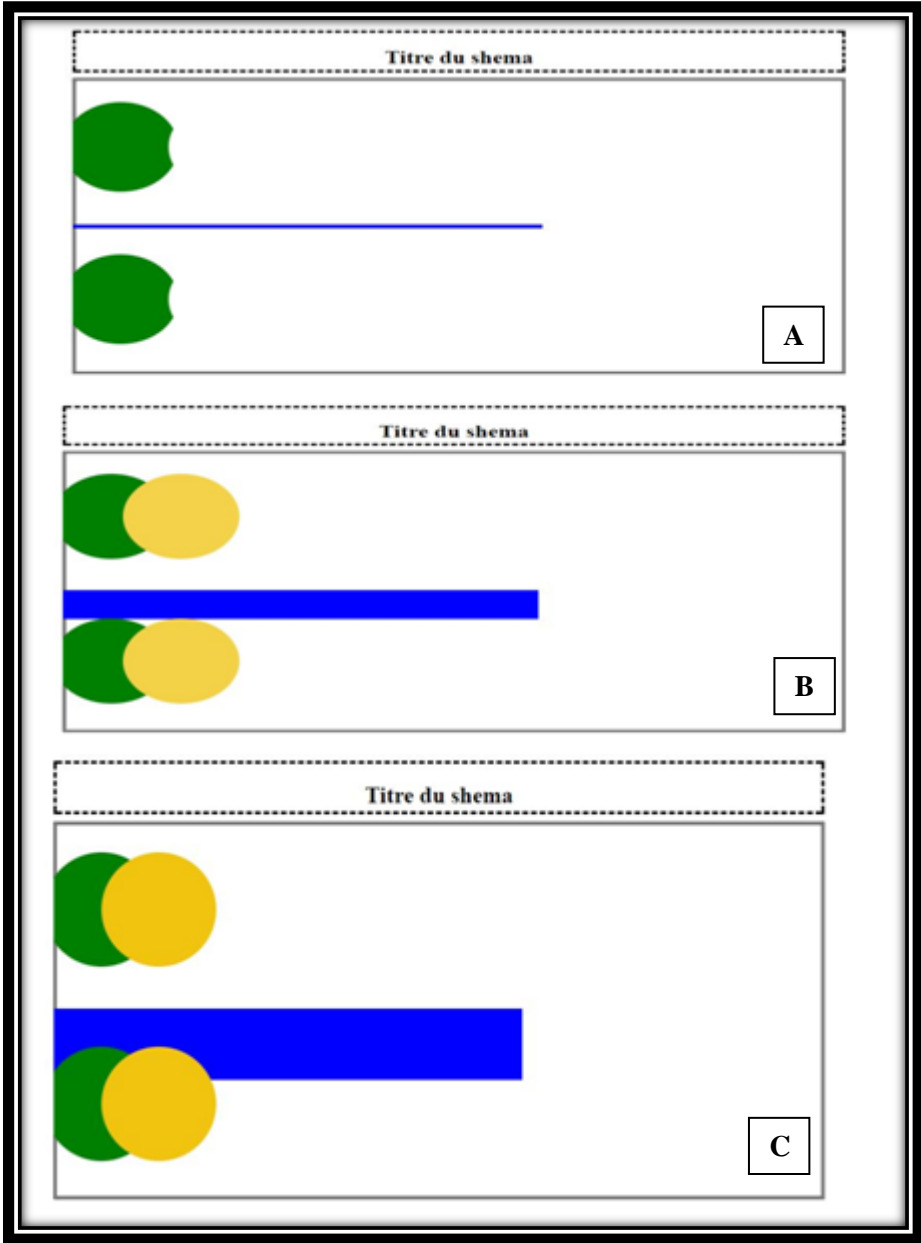


Figure 24:modèle chorématique qui représente un le feu de la forêt et l'inondation

3 Résultat et discussions

3.1 Gestion de projet : Diagramme de Gantt

Le stage a commencé le 22 mai 2017 pour une durée de 8 semaines. Dès le début j'ai adopté ce planning mais il était possible qu'on travaillait une seule tâche plusieurs jours sans respecter le délai imposé. Pourtant, le but n'était pas d'exécuter tous les travaux mais plutôt de gérer avec prudence notre temps afin que le prototype soit opérationnel avant le 13 juillet 2017. Du coup nous avons modifié plusieurs fois le planning mais vous trouverez ci-dessous la version finale (figure 25). Ce planning a été créé avec le logiciel de Ganttproject.

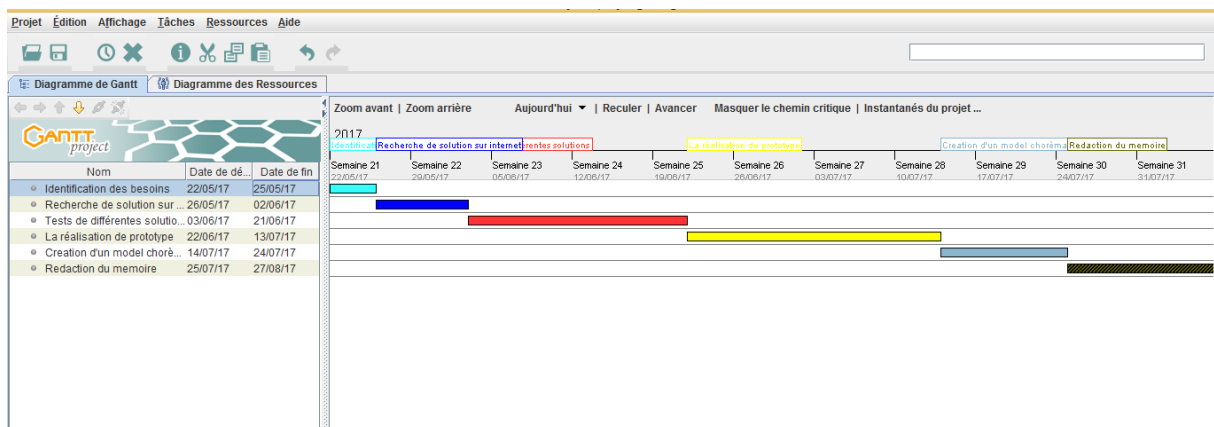


Figure 25: Extrait du diagramme de Gantt

3.2 Description du prototype de carte interactive pour les événements historiques sur le territoire du Haut Vicdessos

3.2.1 Architecture logicielle et solutions techniques retenues

L'architecture de notre application est basée essentiellement sur GeoServer, OpenLayers, PostgreSQL/PostGIS et un serveur web, pour un accès lorsque l'utilisateur tape « localhost » sur le navigateur.

Le serveur web analyse la requête http et envoie une requête WMS-Time au serveur cartographique. Puis, le serveur cartographique récupère les données dans la base PostGIS, en utilisant la requête GetMap pour le serveur WMS. A partir de ces données, le serveur cartographique génère une image correspondant à la requête.

Le serveur web rassemble tous les éléments (fichiers HTML, php, images, scripts, style) de la page web y compris l'image générée par le serveur cartographique et les envoie au navigateur

de l'utilisateur. Le navigateur reçoit et met en forme les différents éléments et affiche la page web.

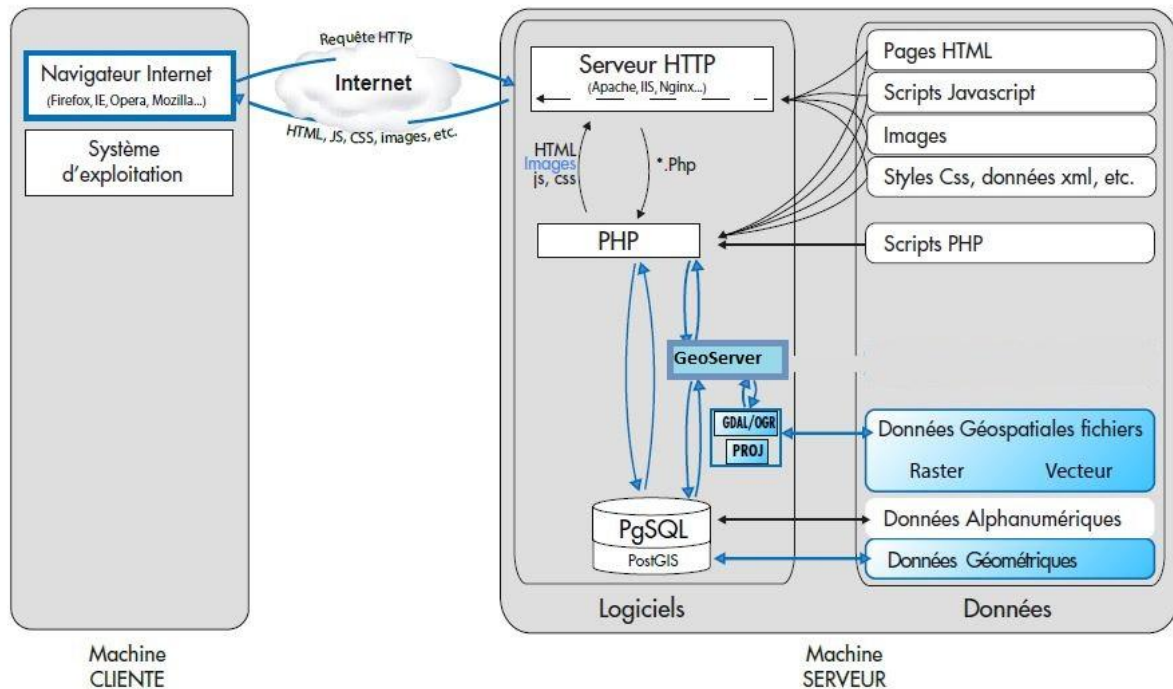


Figure 26: Architecture du prototype. (Source : L. Jégou, cours SIGMA 2016)

3.2.2 Présentation du prototype

Nous sommes passés par différentes étapes pour arriver à la phase finale de ce prototype et certaines étapes prenaient des jours pour passer à l'étape suivante. De manière générale, ce prototype est structuré par trois points importants : une page html, la page php, les différentes fonctions de JavaScript :

- ❖ La page html contient le code source de l'application, elle appelle la requête php et charge les bibliothèques JavaScript.
- ❖ La page php permet d'effectuer une requête php côté serveur pour récupérer les années disponibles dans la base de données afin que l'utilisateur clique l'année qui l'intéresse (figure 28).
- ❖ Dans notre code, il y a plusieurs fonctions JavaScript comme **update** qui permet de mettre à jour les cartes à chaque fois que l'utilisateur clique sur une année (figure 29). Elle permet également de rendre dynamique la demande **GetLegendGraphic**, c'est-à-dire une légende dynamique (figure 32).
- ❖ La fonction **Slider** (figure 27) permet de créer le curseur temporel de notre application, elle récupère également les années disponibles dans la base de données grâce à la requête php, puis elle affiche automatiquement les années sur le curseur (figure 30).

- ❖ La fonction **Control.WMSGetFeatureInfo**, permet de requêter et d'afficher la table attributaire lorsque l'utilisateur clique sur la couche (figure 33 et 31).

L'interface de notre prototype est composée d'une seule partie mais elle est constituée de trois sous-parties :

- la partie la plus haute comporte le curseur temporel qui sélectionne les années disponibles.
- La partie centrale affiche la carte interactive.
- La partie la plus basse affiche la légende de la carte.

```

64
65         slider = new Slider("#ex13", {
66             ticks: annee_client,
67             ticks_labels: annee_client, |
68             ticks_snap_bounds: 300
69         });
70
71         slider.on('change', update_date);
72

```

Figure 27:fonction qui permet de générer le curseur

```

1  <?php
2
3  $conn = pg_connect("host=localhost dbname=bd3_postgis user=postgres password=asmadawo");
4  if (!$conn) {
5      echo "Erreur de connexion.\n";
6      exit;
7  }
8
9  $sql = "SELECT extract(YEAR FROM gestion_eau.annee)as annee_client from gestion_eau
10 union
11 select extract(YEAR FROM ev_hydro3.annee)as annee_client from ev_hydro3
12 union
13 select extract(YEAR FROM date_deb)as annee_client from usine
14 union
15 select extract(YEAR FROM date_fin)as annee_client from usine
16 order by annee_client
17
18 ";
19
20 $requeteSQL = pg_exec($conn, $sql);
21
22 //Génération de la chaîne GeoJSON
23 $str = 'var annee_client =[';
24 while ($result = pg_fetch_array($requeteSQL)) {
25     $str .= $result['annee_client'].',';
26 }
27 $str = substr($str,0,strlen($str)-1);
28 $str .= '];';
29
30 //close($conn);
31
32 echo $str;
33
34 ?>

```

Figure 28:requête php pour récupérer les années disponible sur la base de données

```

110
111     function update_date() {
112
113         var periode = slider.getValue(); |
114         var nvannee = 'annee:' + parseInt(periode);
115         jpl_wms.mergeNewParams({'viewparams':nvannee});
116         document.getElementById('legend').src='http://localhost:8080/geoserver/wms?REQUEST=GetLegendGraphic&VERSION=1.0.0&FORMAT=image/
117             png&WIDTH=20&HEIGHT=20&LAYER=test1:union';
118         var l = document.getElementById('libelle');
119         l.textContent = periode;
120     }

```

Figure 29:fonction update

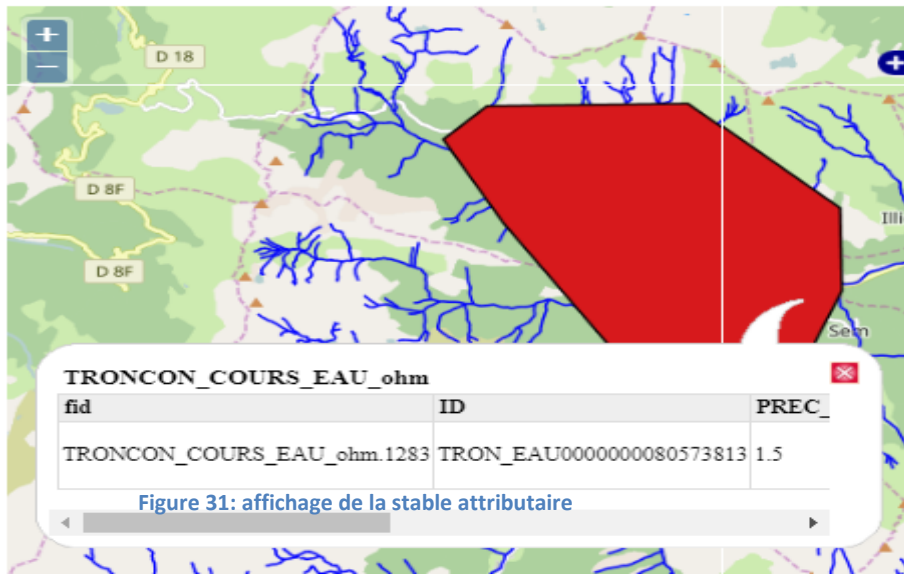


Figure 31: affichage de la stable attributaire

```

info = new OpenLayers.Control.WMSGetFeatureInfo({
  url: 'http://localhost:8080/geoserver/wms',
  title: 'Identify features by clicking',
  queryVisible: true,
  eventlisteners: {
    getfeatureinfo: function(event) {
      map.addPopup(new OpenLayers.Popup.FramedCloud(
        "chicken",
        map.getLonLatFromPixel(event.xy),
        null,
        event.text,
        null,
        true
      ));
    }
  }
});
map.addControl(info);
info.activate();
}

```

Figure 32: fonction Control.WMSGetFeatureInfo

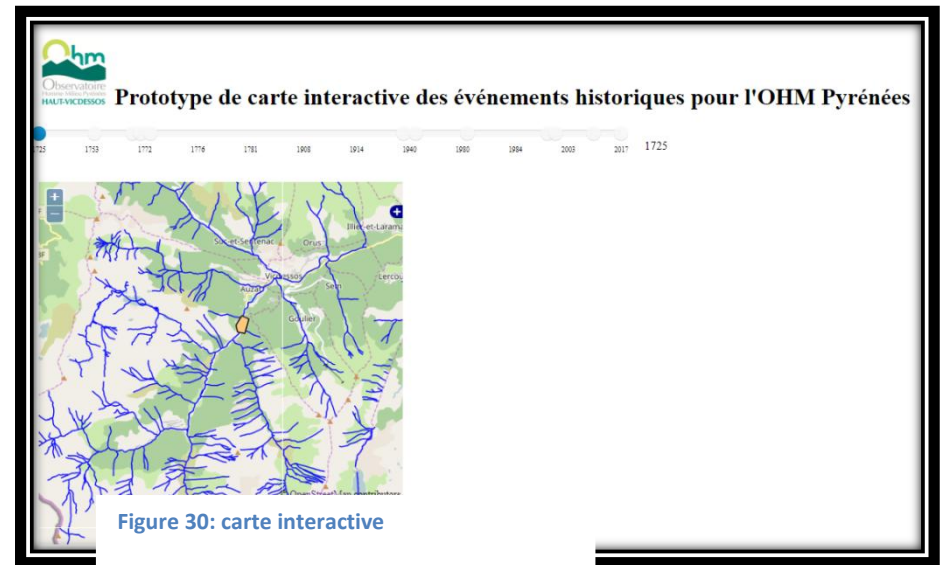


Figure 30: carte interactive

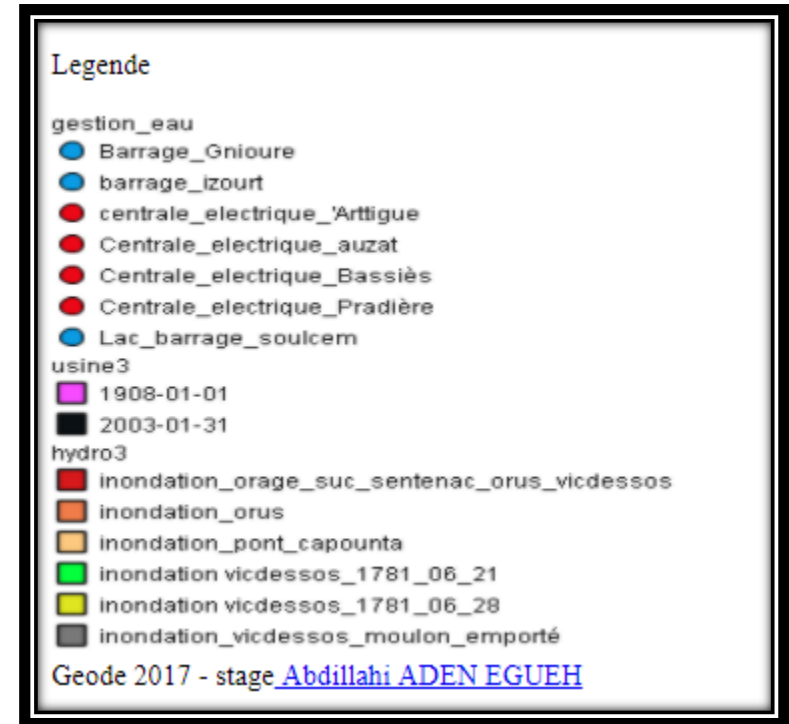


Figure 33: légende dynamique

3.2.3 Discussion

Nous rappelons que le contexte du stage prévoyait de développer avec des logiciels open source. La réalisation d'une telle application est en réalité confrontée à plusieurs contraintes : non seulement la durée de stage était limitée à 8 semaines mais l'objectif était que le prototype soit opérationnel avant l'achèvement de mon stage.

En outre, le protocole WMS-T prend en charge seulement des données qui sont régulières et continues dans le temps comme par exemple celles de météorologie, mais dans notre cas les événements étaient non réguliers dans le temps. Pour contourner ce problème nous avons utilisé le paramètre SQL View de GeoServer pour obtenir les données via une requête temporelle et leur style associé.

Cette application atteint la phase stable mais comme c'est un travail exploratoire, il faut améliorer certaines fonctionnalités comme lorsque l'utilisateur clique une année et que cette année ne correspond à aucun événement. Il faudrait alors que l'application affiche l'événement le plus proche au lieu de carte vide, ce qui reste à développer.

3.2.4 Livraison du prototype au commanditaire

Nous avons développé le prototype avec un serveur local (localhost) afin qu'on puisse transférer les données au serveur du laboratoire GEODE. Il faut que les données soient disponibles sur Internet pour que les membres des chercheurs de laboratoire GEODE puissent consulter les cartes interactives.

Pour cela, nous avons transféré le prototype au commanditaire (Madame Émilie LERIGOLEUR) en donnant un répertoire qui contient les différents fichiers et scripts tels que la page html, php ainsi que les deux requêtes SQLView de GeoServer.

Ensuite la commanditaire présentera ces projets à la direction du laboratoire GEODE, si le travail reçoit un avis favorable, elle continuera à développer le prototype.

3.3 Présentation du modèle chorématique

Nous rappelons que l'objectif était d'abord de tester des chorèmes animés à partir de la bibliothèque D3.js, ensuite de vérifier les diverses fonctionnalités que D3.js nous offre pour modéliser tout type de forme.

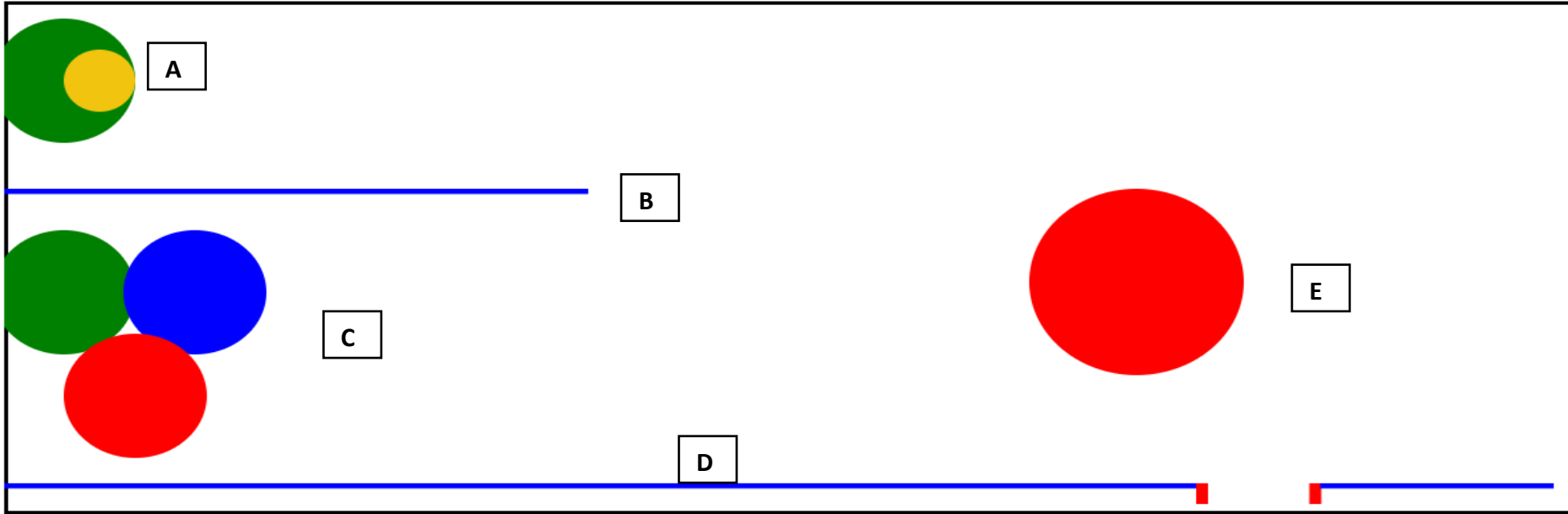
Pour cela nous avons testé plusieurs chorèmes simples en commençant par le modèle de l'usine (figure 23) ensuite nous sommes allés plus loin en essayant de modéliser deux événements en même temps. C'est l'exemple de l'incendie forestier et de l'inondation, en essayant de modéliser d'une part l'inondation qui ravageait la forêt et d'autre part on schématisait la propagation de l'incendie forestier (figure 24). On s'est alors rendu compte qu'il est possible de représenter d'une autre manière la propagation du feu, plus convenable que celle d'avant.

Pour cela, nous avons essayé de schématiser un modèle chorématique qui comporte cette fois-ci plusieurs événements simultanément. Nous avons choisi de représenter la propagation des incendies forestiers par un cercle jaune qui se développe sur un autre cercle, vert, cela traduisant plus ou moins la propagation du feu (figure 34_A).

Pour l'association des cercles qui deviennent un seul cercle bleu, ce schéma traduit les regroupements des communautés de communes (figure 34_C). Pour le schéma de l'inondation nous n'avons pas changé sa forme mais cette fois il va ravager les communautés de communes lorsque la taille de l'épaisseur augmente (figure 34_B). Pour l'événement usine nous avons gardé le même schéma qu'avant c'est-à-dire le rayon du cercle est fixé mais il apparaît et disparaît puis il change la couleur (figure 34_E). La ligne bleue schématise un phénomène qui est continu dans le temps et qui s'arrête un laps de temps puis reprend sa continuité (figure 34_D), cet événement n'est pas mentionné dans le tableau d'événement (Annexe 3), car il s'agit d'un ajout propre, conçu et modélisé.

La bibliothèque D3 nous offre des fonctionnalités diverses pour adapter la représentation selon le besoin et paraît surtout utile pour les chorèmes animés. Le plus grand inconvénient est le temps d'apprentissage du langage de D3, surtout pour le débutant en programmation. Une personne qui connaît la programmation appréciera la puissance de d3.js. Ainsi, pour concevoir des modèles chorématiques animés, on peut passer la bibliothèque D3 mais cela nécessite une bonne connaissance de la programmation.

Schématisation d'un model chorématique



Schématisation d'un model chorématique

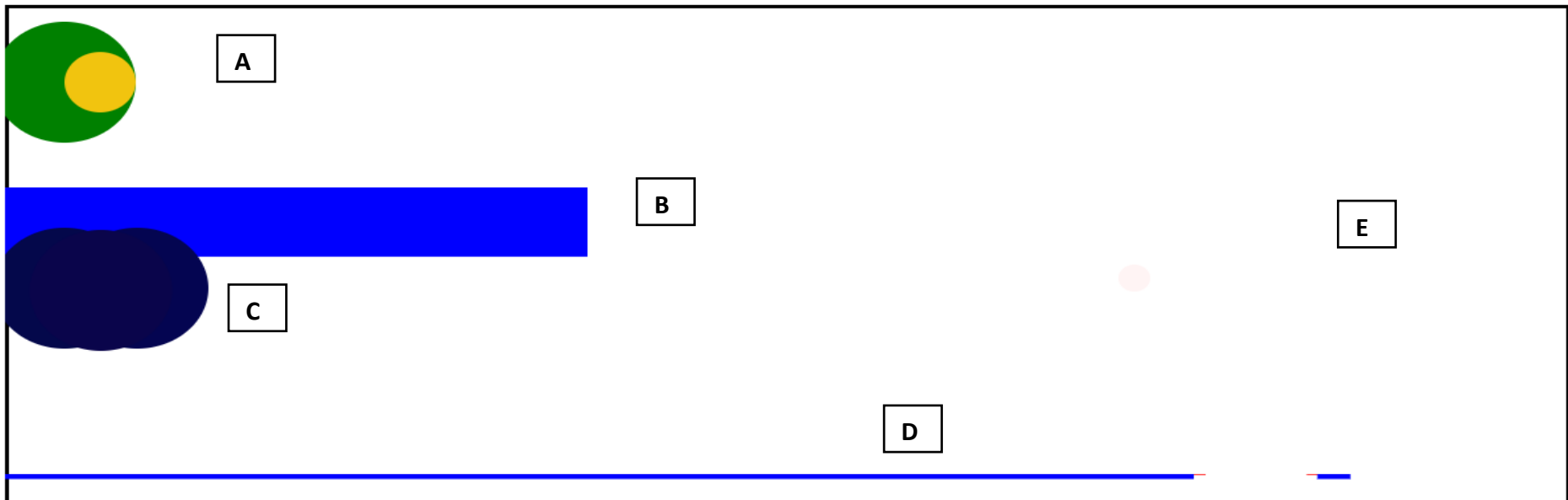


Figure 34: Schématisation d'un modèle chorématique

4 Conclusion

Le prototype réalisé durant ce stage est une application qui permet aux chercheurs de faciliter leurs recherches. Cette application a été développée à partir de logiciels Open Source. Elle permet la géovisualisation de l'évolution dans le temps du paysage, avec l'affichage des tables attributaires. L'utilisateur a la possibilité de choisir une année et le prototype affiche tous les événements correspondant à cette date. Le prototype que j'ai développé répond aux principaux objectifs fixés en début de stage. L'application fonctionne de manière très satisfaisante mais c'est un travail exploratoire que Mme Émilie Lerigoleur va continuer si le travail reçoit un avis positif. Dans la partie chorématique, j'ai exploité la puissance de la bibliothèque D3 en JavaScript avec laquelle j'ai testé plusieurs exemples de chorèmes animés simples. Malgré le fait que la durée de stage fut de 8 semaines, j'ai eu la chance de tester un modèle chorématique qui est complexe mais aussi animé.

J'ai appris à travailler en toute autonomie et la responsabilité de gérer un projet tout en respectant la durée fixée du stage, malgré les difficultés rencontrées, que j'ai réussies à surmonter. Sur le plan personnel, c'est vraiment difficile de résumer en quelques lignes l'expérience enrichissante que ce stage m'a apportée tant au niveau professionnel, relationnel que social. Je me suis également familiarisé avec la communication dans le monde professionnel, d'ailleurs j'ai apprécié de travailler auprès des chercheurs au sein des laboratoires. Ce stage m'a permis également d'approfondir mes compétences en webmapping, particulièrement en programmation web surtout en PHP et JavaScript. J'ai créé une base de données sous PostgreSQL/PostGIS, j'ai pu mettre en place un serveur cartographique avec GeoServer, puis j'ai testé plusieurs API de Webmapping, notamment OpenLayers et Leaflet. J'ai aussi appris la manière de repérer les erreurs générées sur la console de Chrome et de Mozilla pour mieux comprendre le fonctionnement des scripts et corriger les erreurs.

Bibliographie – webographie

R. Brunet, la carte de modèles et les chorèmes, M@ppemonde 84/6

Jean-Paul Cheylan, Les processus spatio-temporels: quelques notions et concepts préalables à leur représentation, M@ppemonde 87 (2007.3).

Lahouari Kaddouri, Réflexion sur la sémiologie graphique animée des flux, M@ppemonde 89 (2008.1).

Gbati Batame TAFAMBA-DABOU, prototypage d'une application websig de reseau d'alimentation en eau potable, rapport de stage M2 SIGMA 2013.

Jean Claude OUEDRAOGO, Mise en place d'une interface webmapping sur la «Capitalisation des expériences de gestion durable de la fertilité des sols au Burkina Faso», rapport de stage M2 IASIG.

Site OHM : http://w3.ohmpyr.univ-tlse2.fr/presentation_ohm_pyr.php).

Test des chorèmes animé : <https://bl.ocks.org/ljegou>

Time Support in GeoServer WMS:

<http://docs.geoserver.org/stable/en/user/services/wms/time.html>

Exemple WMS Time: <http://dev.openlayers.org/examples/wmst.html>

SQL Views : <http://docs.geoserver.org/stable/en/user/data/database/sqlview.html>

contrôle WMSGetFeatureInfo : <http://dev.openlayers.org/examples/getfeatureinfo-popup.html>

Slider Bootstrap : <http://seyria.com/bootstrap-slider/>

QGIS Time-manager : <https://anitagraser.com/projects/time-manager/>

OpenLayer : <https://openlayers.org/>

D3.js : <https://d3js.org/>

Leaflet.TimeDimension : <https://github.com/socib/Leaflet.TimeDimension>

Leaflet.wms.js : <https://github.com/ev4debug/leaflet.wms>

Leaflet TimeDimension Exemples: <http://apps.socib.es/Leaflet.TimeDimension/examples/>
(exemple leaflet)

Leaflet.timeline : <https://github.com/skeate/Leaflet.timeline>

Table de matière

1	Introduction.....	1
1.1	Les laboratoires d'accueil : l'unité mixte de recherche Géographie de l'environnement (GEODE) et le Laboratoire Interdisciplinaire Solidarités, Sociétés, Territoires (LISST).....	2
1.2	L'Observatoire Hommes-Milieus Pyrénées Haut Vicdessos et Haute vallée du Gave.....	2
1.3	Objectifs du stage.....	5
1.3.1	Prototype de carte interactive pour les événements historiques sur le territoire du Haut Vicdessos	5
1.3.2	Chorèmes.....	5
1.4	Démarche méthodologique globale.....	5
2	Présentation du travail technique, des problèmes et de leur résolution	7
2.1	Objectif "prototype de carte interactive pour l'OHM	7
2.1.1	Identification des besoins.....	7
2.1.2	Recherche de solutions	8
2.1.3	Première série de tests pour orienter le choix des solutions.....	9
2.1.4	Installation des logiciels GeoServer et PostgreSQL	11
2.1.5	Création d'une base de données PostgreSQL/PostGIS	14
2.1.6	Création de requêtes "SQL View" dans GeoServer	16
2.1.7	Tests de différentes solutions	18
2.1.7.1	Test 1 : Gestion des styles d'affichage	18
2.1.7.2	Test2 : ajout d'un curseur temporel.....	19
2.2	Objectif "chorèmes"	22
2.2.1	Identification des besoins.....	22
2.2.2	Recherche de solution	24
2.2.3	Tests de différentes solutions	25
2.2.3.1	Test 1 : représentation des chorème simple.....	25
3	Résultat et discussions	27
3.1	Gestion de projet : Diagramme de Gantt.....	27
3.2	Développement du prototype de carte interactive pour les événements historiques sur le territoire du Haut Vicdessos.....	27
3.2.1	Architecture logicielle et solutions techniques Retenues	27
3.2.2	Présentation du prototype	28
3.2.3	Discussion	31

3.2.4	Livraison du prototype au commanditaire	31
3.3	Présentation du modèle chorématique	32
4	Conclusion	34

LISTE DE FIGURE

Figure 1:Utilisation de la Loi du Quintilien pour affiner les besoins	8
Figure 2:Configuration de dimension TIME pour une couche WMS.....	10
Figure 3:Script du test 1 avec Leaflet	10
Figure 4: Résultat du premier test avec Leaflet	10
Figure 5:définition du nom de l'utilisateur et son mode de passe pour l'administrateur	11
Figure 6:Définition de port de GeoServer	12
Figure 7:: Installation de GeoServer en tant que service	12
Figure 8:installation des binaires	13
Figure 9:requête d'activation de l'extension PostGIS	13
Figure 10: Interface de PgAdminIII.....	14
Figure 11:création de basse de donnée « bdd3_postgis » sur l'interface graphique de pgAdminIII....	15
Figure 12:Importation d'une couche shapefile	15
Figure 13:Configuration d'une requête SLQ View d'un événement ponctuelle	16
Figure 14: Configuration d'une requête SLQ View d'un événement continue	17
Figure 15: Configuration de style SLD dans le GeoServer	18
Figure 16:Affectation des styles dans GeoServer.....	19
Figure 17:code source du curseur html.....	20
Figure 18:curseur illisible dans le navigateur Mozilla	20
Figure 19:curseur compatible au navigateur chrome	21
Figure 20:Curseur compatible au navigateur Mozilla	21
Figure 21:la table de 28 chorème de bases. Source (R.Brunet, 1986)	23
Figure 22:Style de représentation des événements	24
Figure 23:Modélisation de l'événement « ouverture et fermeture de l'usine »	25
Figure 24:modèle chorématique qui représente un le feu de la forêt et l'inondation.....	26
Figure 25:Extrait de diagramme de Gantt.....	27
Figure 26:Architecture du prototype. (Source : L. Jégou, cours SIGMA 2016).....	28
Figure 27:fonction qui permet de générer le curseur	29
Figure 28:requête php pour récupérer les années disponible sur la base de données.....	29
Figure 29:fonction update.....	29
Figure 31: affichage de la stable attributaire	30
Figure 30: carte interactive	30
Figure 32: légende dynamique.....	30
Figure 33:fonction Control.WMSGetFeatureInfo.....	30
Figure 34: Schématisation d'un modèle chorématique	33

Annexe

Annexe 1 : code source du prototype

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

    <style>
      .smallmap {

        width: 500px; height : 500px;

      }
      #legend {
        width: 350px; height: 300px;
      }

.slider {
  width: 800px !important;
}

.slider-tick-label{

  font-size: 0.60em;
  margin-top: 20px;
}

    </style>
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <title>OpenLayers: WMS + Time</title>
    <link rel="stylesheet" href="http://localhost/wms/bootstrap-
slider.min.css" type="text/css">
    <script src="http://localhost/wms/bootstrap-slider.min.js"></script>
    <script src="http://localhost/wms/get_data.php"></script>
    <link rel="stylesheet"
href="http://localhost/wms/theme/default/style.css" type="text/css">
    <script src="http://localhost/wms/OpenLayers.js"></script>

    <script type="text/javascript">
      var map, jpl_wms, slider,info;
      function init(){
        map = new OpenLayers.Map({
          div: "map",
          projection: "EPSG:3857"
        });

        var osm = new OpenLayers.Layer.OSM();
```

```

var ol_wms = new OpenLayers.Layer.WMS( "OpenLayers WMS",
    "http://vmap0.tiles.osgeo.org/wms/vmap0?", {layers:
'basic'} );
eau_wms = new OpenLayers.Layer.WMS( "cour d'eau",
    "http://localhost:8080/geoserver/wms",
    {layers:
'TRONCON_COURS_EAU_ohm',transparent:true,format:'image/jpeg'});

    // "AIzaSyA5Y2liG-TaMn_TvTDbVdckmrmw-h5PlN0");
jpl_wms = new OpenLayers.Layer.WMS( "OHM Vicdessos foret",
    "http://localhost:8080/geoserver/wms",
    {layers:
'hydro3,gestion_eau,usine3',transparent:true,format:'image/jpeg',
viewparams: 'annee:1725'});

    slider = new Slider("#ex13", {
    ticks: annee_client,
    ticks_labels: annee_client,
    ticks_snap_bounds: 300
    });

    slider.on('change', update_date);

map.addLayers([osm, ol_wms, eau_wms, jpl_wms]);
map.addControl(new OpenLayers.Control.LayerSwitcher());
map.zoomToExtent(new
OpenLayers.Bounds(1.40,42.79,1.53,42.69).transform("EPSG:4326",
"EPSG:3857"));

    info = new OpenLayers.Control.WMSGetFeatureInfo({
    url: 'http://localhost:8080/geoserver/wms',
    title: 'Identify features by clicking',
    queryVisible: true,
    eventListeners: {
        getfeatureinfo: function(event) {
            map.addPopup(new OpenLayers.Popup.FramedCloud(
                "chicken",
                map.getLonLatFromPixel(event.xy),
                null,
                event.text,
                null,
                true
            ));
        }
    }
    });
map.addControl(info);
info.activate();

}

```


Annexe2: Code source du chorème

```
<!DOCTYPE html>
<meta charset="utf-8">
<head>

<style type="text/css">

#contour{

    width: 1310px;
    height: 50px;
    border: 3px blue dashed;
    border-color: black;
    padding-top: -320px;
    text-align: center;
}
#sh{
    border: 3px black double;
width: 1310px;
    height: 490px;

    margin-top: -750px;
}

</style>
<script src="http://d3js.org/d3.v3.js"></script>
</head>

<body>

<div id="contour"> <h2> Schématisation d'un model chorématique </h2></div>
<script id="sc">

var canvas=d3.select("body")
    .append("svg")
    .attr("width",1000)
    .attr("height",200);

canvas.style.border='3px blue dashed';

var canva=d3.select("body")
    .append("svg")
    .attr("width",300)
    .attr("height",200);
var canvais=d3.select("body")
    .append("svg")
    .attr("width",1300)
    .attr("height",280);

var circle1 = canvas.append("circle")
    .attr("cx", 50)
    .attr("cy", 100)
    .attr("r", 60)
    .style("fill", "green");
```

```
var circle2 = canvas.append("circle")
    .attr("cx", 80)
    .attr("cy", 100)
    .attr("r", 0)
    .style("fill", "#f1c40f");
```

```
    circle2.transition()
    .duration(10500)
    .delay(1000)
    .attr("r", 30)
    .style("fill", "#f1c40f")
    //deuxime fois
    .transition()
    .duration(10500)
    .attr("r", 0)
    .style("fill", "#f1c40f")
    .transition()
    .duration(10500)
    .attr("r", 30)
    .style("fill", "#f1c40f");
```

```
var line=canvais.append("line")
    .attr("x1",0)
    .attr("x2",490)
    .attr("stroke","blue")
    .attr("stroke-width",10);
```

```
    line.transition()
    .duration(9500)
    .attr("stroke-width",150)
    .transition()
    .duration(9500)
    .attr("stroke-width",10);
```

```
var circle3 = canvais.append("circle")
    .attr("cx", 50)
    .attr("cy", 100)
    .attr("r", 60)
    .style("fill", "green");
```

```
var circle4 = canvais.append("circle")
```

```

        .attr("cx", 160)
        .attr("cy", 100)
        .attr("r", 60)
        .style("fill", "#f1c40f");

var circle5 = canvais.append("circle")
    .attr("cx", 110)
    .attr("cy", 200)
    .attr("r", 60)
    .style("fill", "red");

circle3.transition()
    .duration(9500)
    .delay(1000)
    .style("fill", "#04054D")
    .transition()
    .duration(9500)
    .style("fill", "green")
    .transition()
    .duration(9500)
    .style("fill", "#04054D")
    .transition()
    .duration(9500)
    .style("fill", "green")
    .transition()
    .duration(9500)
    .style("fill", "#04054D")
    .transition()
    .duration(9500)
    .style("fill", "green");

circle5.transition()
    .duration(9500)
    .delay(1000)
    .attr("cx", 80)
    .attr("cy", 100)
    .style("fill", "#04054D")
    .transition()
    .duration(9500)
    .attr("cx", 110)
    .attr("cy", 200)
    .style("fill", "red")
    .transition()
    .duration(9500)
    .attr("cx", 80)
    .attr("cy", 100)
    .style("fill", "#04054D")
    .transition()
    .duration(9500)
    .attr("cx", 110)
    .attr("cy", 200)
    .style("fill", "red");

```

```

circle4.transition()
  .duration(9500)
  .delay(1000)
  .attr("cx", 110)
  .attr("cy", 100)
  .style("fill", "#04054D")
  .transition()
  .duration(9500)
  .attr("cx", 160)
  .attr("cy", 100)
  .style("fill", "blue")
  .transition()
  .duration(9500)
  .attr("cx", 110)
  .attr("cy", 100)
  .style("fill", "#04054D")
  .transition()
  .duration(9500)
  .attr("cx", 160)
  .attr("cy", 100)
  .style("fill", "blue");

var circle= canvais.append("circle")
  .attr("cx", 950)
  .attr("cy", 90)
  .attr("r", 90)
  .attr("fill", "blue");

// Variation de la couleur

circle.transition()
  .duration(10000)
  .attr("fill", "white")
  .attr("r", 10)
  .transition()
  .duration(10000)
  .attr("fill", "red")
  .attr("r", 90);

var canvas=d3.select("body")
  .append("svg")
  .attr("width", 1300)
  .attr("height", 280);

var lines=canvas.append("line")
  .attr("x1", 1005)
  .attr("y1", -10)
  .attr("x2", 1005)
  .attr("y2", 0)
  .attr("stroke", "red")
  .attr("stroke-width", 10);

```



```

lines.transition()
  .duration(9500)
  .delay(10000)
  .attr("y2",20);

var liness=canvas.append("line")
  .attr("x1",1100)
  .attr("y1",-10)
  .attr("x2",1100)
  .attr("y2",0)
  .attr("stroke","red")
  .attr("stroke-width",10);

liness.transition()
  .duration(9500)
  .delay(10000)
  .attr("y2",20);

var line=canvas.append("line")
  .attr("x1",0)
  .attr("x2",20)
  .attr("stroke","blue")
  .attr("stroke-width",10);

line.transition()
  .duration(9500)
  .delay(1000)
  .attr("x2",1000);

var ligne=canvas.append("line")
  .attr("x1",1104)
  .attr("x2",1104)
  .attr("stroke","blue")
  .attr("stroke-width",10);

ligne.transition()
  .duration(9600)
  .delay(10100)
  .attr("x2",1800);

</script>
<div id="sh"></div>

<div id="bas"></div>

</body>

```