

# Editeur graphique d'automate fini

2015 - 2016

Rapport de conception



Université Claude Bernard



Lyon 1

# Rapport de conception

## Sommaire

Présentation générale du projet.....	3
Le contexte .....	3
Le projet .....	3
L'équipe .....	3
Le planning .....	4
Bête à cornes.....	5
Diagramme FAST .....	6
Cadre de réponse .....	7
Technologique.....	7
JavaScript .....	7
D3.js .....	7
JQuery.....	7
HTML .....	8
CSS .....	8
Solution .....	9
Rendu.....	10
Application .....	10
Utilisation générale .....	10
Raccourcis.....	10
Boutons et panneau latéraux .....	11
Améliorations .....	11
Expériences et difficultés .....	12
Difficultés rencontrées .....	12
Changements depuis la première présentation .....	13
Conclusion.....	13
Perspectives pour l'éditeur graphique .....	13
Apports personnels .....	13
Annexes .....	15
Webographie.....	15
Dépôt GitHub .....	15
Table des illustrations .....	15
Remerciements .....	16

# Rapport de conception

# Rapport de conception

## Présentation générale du projet

### Le contexte

Via des automates finis, Monsieur Champin donne la possibilité à des acteurs non informaticiens de transformer des données de bas niveaux, vers des données de plus haut niveaux. Par exemple, un psychologue qui étudie la façon dont une personne conduit un véhicule, recevra des données de bas niveaux telles que : tourner le volant de 30° dans le sens horaire, appuyer sur l'accélérateur, etc... Mais la combinaison de ces informations de bas niveaux peut permettre de générer une information particulière, par exemple : un dépassement, le passage d'un cassis, etc...

### Le projet

L'éditeur graphique d'automate fini a pour vocation de permettre à des individus, a priori sans connaissance informatique, de modéliser des automates finis, sans connaître de langage de programmation. Ce projet est sous-jacent au travail de Monsieur Champin : le moteur d'automate fini. Le projet suit donc les règles imposées par la réalisation de Monsieur Champin. L'éditeur s'inscrit dans un cadre logiciel destiné à simplifier la création, l'édition et la visualisation de ce type d'automates.

### L'équipe

Monsieur Pierre-Antoine CHAMPIN est le manager du projet (MOA), ainsi que le commanditaire. Messieurs Christophe BADET, Martin BOLOT, Rémy FISCHER et Jean-Sébastien FONTA sont les techniciens réalisant le projet (MOE).

# Rapport de conception

## Le planning

Nous avons difficilement pu mettre en place un planning strict. Cependant, nous sommes parvenus à fixer des jalons et des périodes approximatives pour chaque activité. Voici un rendu approximatif et non-exhaustif du planning :

Activité	Date de fin
Début	10/2015
Définition, réflexion et choix technologiques	15/12/2015
Conception et modélisation	15/01/2016
Développement	20/05/2016
Débogage	25/05/2016
Intégration	25/05/2016
Rendu final	06/2016

*Figure 1 - Planning*

# Rapport de conception

Expression fonctionnelle du besoin

Bête à cornes

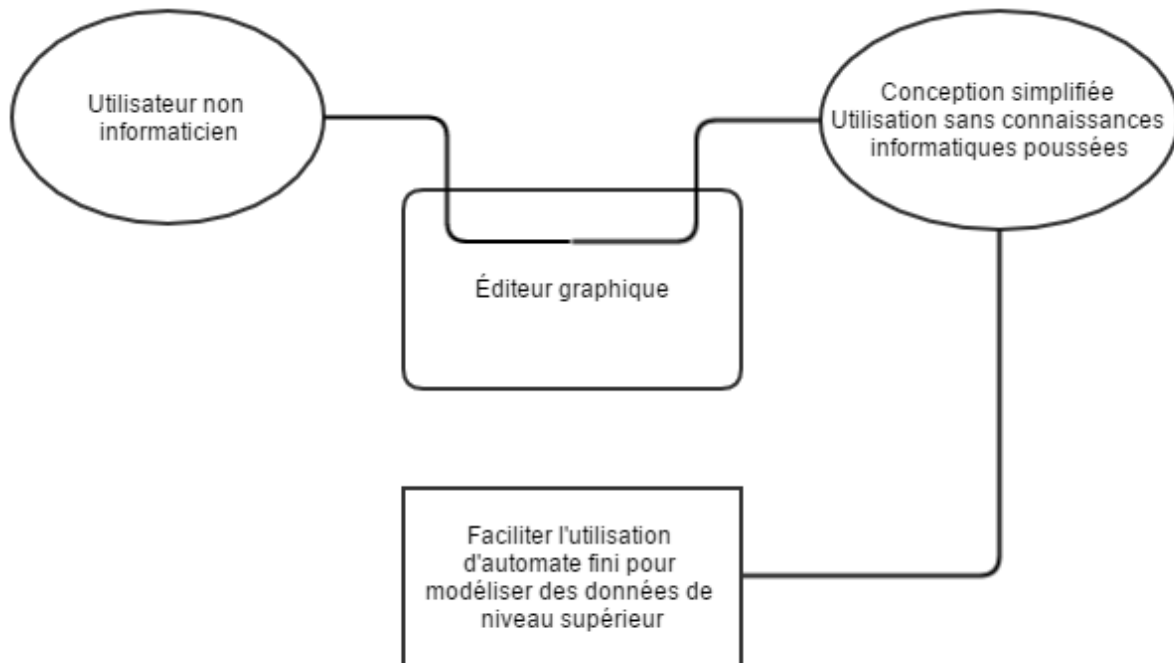


Figure 2 - Bête à cornes

Sur ce diagramme, sont représentés : l'utilisateur, le projet (éditeur), ce sur quoi le projet agit et enfin le résultat attendu.

# Rapport de conception

## Diagramme FAST

Ce diagramme nous permet de synthétiser toutes les fonctions techniques et de services à réaliser pour ce projet.

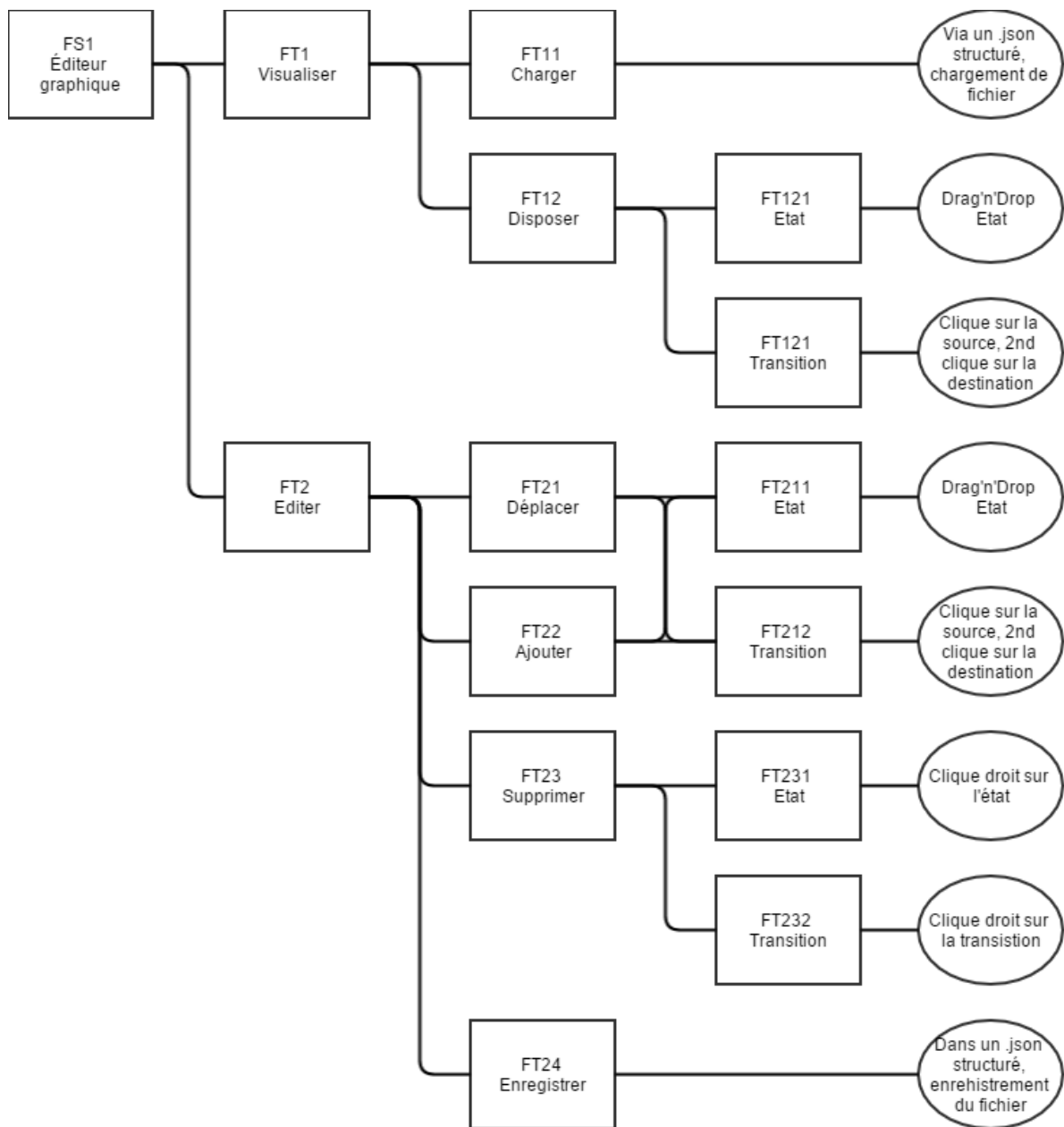


Figure 3 - Diagramme FAST

# Rapport de conception

## Cadre de réponse

### Technologique

*Nous avons utilisé différentes composantes pour mener à bien ce projet*

#### JavaScript

JavaScript est un langage de script léger, orienté objet, principalement connu comme étant le langage de script des pages Web. Il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs Web. C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer de manière fonctionnelle, impérative ou orientée objet. Notre but étant de créer un outil interactif, l'utilisation de ce langage était parfaitement en adéquation avec notre projet puisque le JavaScript permet de rendre les pages Web interactives. Son utilisation faisait partie des prérequis du projet mis en place par Monsieur Champin.

#### D3.js

D3.js est une librairie JavaScript permettant notamment la visualisation de données en permettant la création de graphiques ou de documents dynamiques. D3 permet de faire le liant entre une page Web et un jeu de données pour arriver à produire des visualisations. Nous avons choisi cette librairie, sur les conseils de Monsieur Champin, car c'est l'une des plus populaires en ce qui concerne la représentation graphique de données et parce qu'elle permet de créer facilement des interfaces HTML à partir de données JavaScript ou JSON.

#### jQuery

La devise de jQuery est « write less, do more ». C'est une librairie libre de Javascript sortie en mars 2006. Elle est aujourd'hui maintenue et mise à jour par la jQuery Team. Comme sa devise l'indique, ce framework permet notamment de :

- Parcourir et manipuler le DOM (l'arbre des éléments HTML)
- Gérer des événements provenant de l'utilisateur (clic, survol...)
- Ajouter des effets et animations visuelles
- Gérer des interactions Ajax

La communauté qui l'utilise est importante aujourd'hui, ce qui permet d'accéder à de nombreux supports, documentations, tutoriaux et plugins. Nous l'avons choisie pour



# Rapport de conception

faciliter les manipulations du document HTML n'ayant pas de rapport avec la manipulation de données.

## HTML

HTML est utilisé pour créer et représenter visuellement une page Web. C'est ce qui détermine le contenu et la structure sémantique d'une page Web. Nous nous en sommes servi pour construire notre application notamment via des balises SVG.

## CSS

CSS est un langage de feuille de style en cascade utilisé afin de décrire la présentation d'un document écrit en HTML ou en XML. CSS décrit la façon dont les éléments doivent être affichés à l'écran.

Il faut noter que nous avons démarré notre projet tuteuré sans avoir commencé le module de Web dans le cadre de notre année spéciale. Il a donc fallu pour certains d'entre nous se former depuis la base soit le HTML et CSS, pour ensuite apprendre le Javascript et le jQuery. Les cours disponibles en ligne transmis par Monsieur Champin ainsi que le site Openclassroom nous ont permis de combler ce manque afin de pouvoir avancer dans le projet.

## Solution

Voici un diagramme représentant les diverses fonctionnalités de l'application :

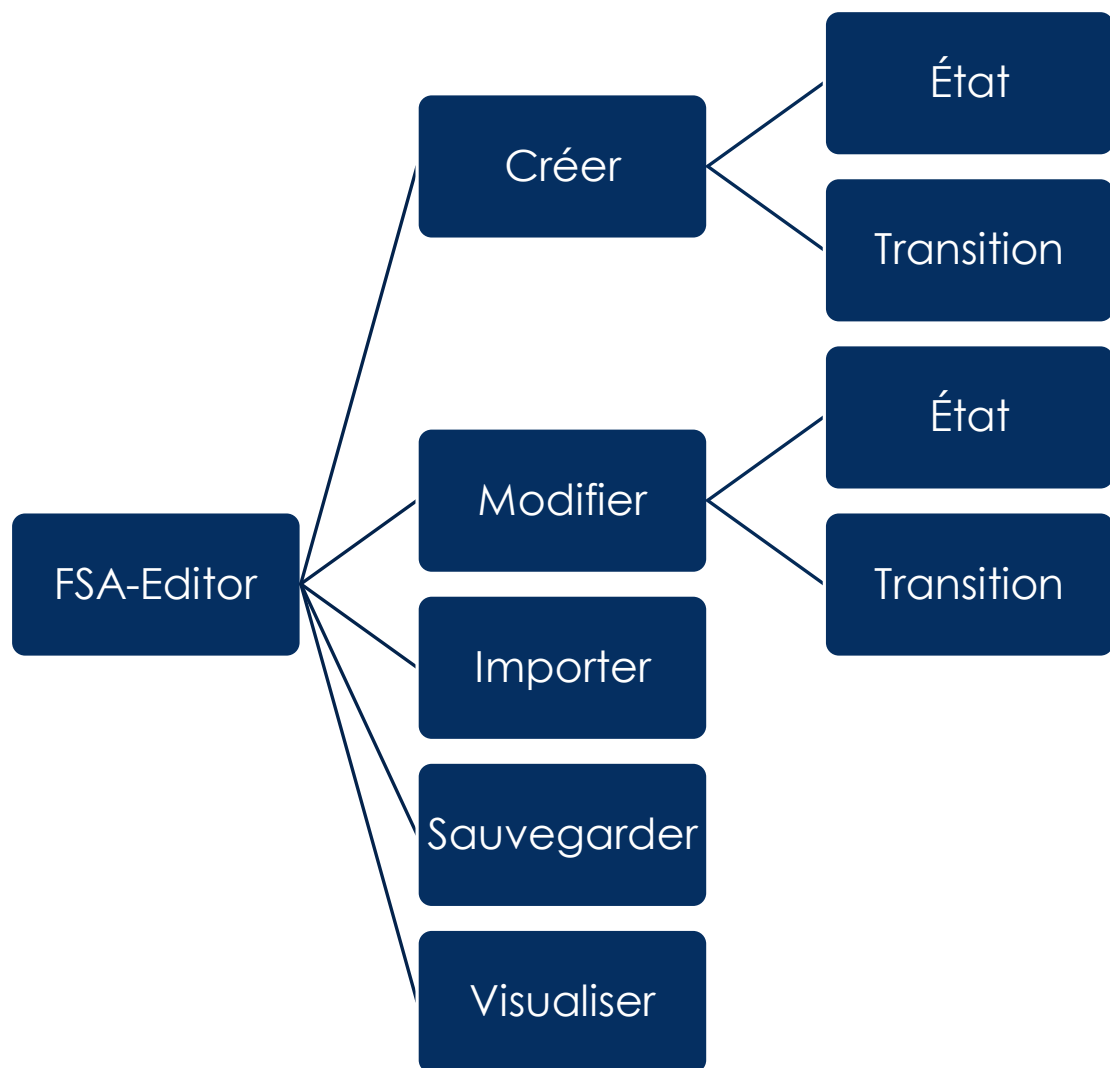


Figure 4 - Schéma de l'application

# Rapport de conception

## Rendu

### Application

#### Utilisation générale

L'application propose au chargement une représentation graphique des différents éléments qui composent l'automate passé en paramètre de la fonction

« fsa\_editor » produite par l'équipe de développement :

- Les états sont représentés par des cercles
- Le nom d'un état est représenté par un texte accolé au cercle le représentant
- Le bruit maximum, optionnel, est représenté par un chiffre entre crochets accolé au nom d'un état
- Les transitions entre états sont représentées par des flèches
- Les conditions de transition sont représentées par du texte accolé au flèches

L'utilisateur peut sélectionner un état en effectuant un clic simple sur le cercle le représentant, entraînant un changement de couleur pour ce dernier. Appuyer sur la touche « ECHAP » aura pour effet d'annuler la sélection d'un premier état. Si l'utilisateur clique sur un état alors qu'un autre état était précédemment sélectionné, une transition va se créer entre les deux états si l'utilisateur renseigne une condition valide, c'est-à-dire ne contenant pas de virgules.

#### Raccourcis

Différents moyens d'interaction sont mis à disposition de l'utilisateur si le mot-clé « edit » a été passé en paramètre de la fonction d'édition :

- Sélectionner un état puis appuyer sur « SUPPR » va permettre de le supprimer
- Sélectionner un état puis appuyer sur « E » va permettre d'afficher un formulaire pour éditer l'ensemble de ses propriétés
- Cliquer sur le nom d'un état permet d'éditer son nom seul

# Rapport de conception

- Cliquer sur le bruit maximum « max\_noise » d'un état permet d'éditer son bruit maximal seul
- Cliquer sur un texte de transition permet d'éditer ou de supprimer la ou les conditions pour cette transition

Le raccourcis « CTRL+Z » et « CTRL+Y » permettent, respectivement, de revenir en arrière ou d'avancer vers l'état, respectivement, précédent ou suivant de l'application. L'état de l'application est enregistré à chaque modification d'un état de l'automate, ou d'une transition dans une limite de 50 retours en arrière possibles.

## Boutons et panneau latéraux

Enfin, trois boutons sont présents dans l'interface, ainsi que deux panneaux latéraux. Le bouton situé en haut de l'interface permet d'afficher un formulaire afin d'éditer les propriétés globales de l'automate.

Les boutons en bas de l'interface servent respectivement à réinitialiser l'automate par rapport au moment de l'ouverture de l'application, et à sauvegarder l'automate sous la forme d'un objet JSON dans le cas où une URL a été fournie en paramètre de la fonction « fsa\_editor ». Dans ce cas, l'objet JSON généré sera envoyé à l'URL spécifiée, sous la forme d'une requête http de type POST.

Le panneau latéral de gauche permet d'afficher un état neutre amovible, qui créera un nouvel état pour l'automate s'il est déposé sur celui-ci. Son fonctionnement n'est pour le moment compatible qu'avec le navigateur Google Chrome.

Le panneau latéral droit permet de visualiser sous forme textuelle l'objet JSON représentant l'automate, mis à jour à chaque modification et indenté pour faciliter la lecture.

## Améliorations

Bien que beaucoup de choses aient pu être réalisées durant ce projet, certaines améliorations potentielles peuvent être envisagées. Ainsi nous avons pour ambition de proposer un ajout dynamique des différents attributs personnalisés pour l'automate ainsi que ses états. Il serait également intéressant de permettre une customisation, par exemple avoir des options visuelles fournies comme la

# Rapport de conception

modification des couleurs de l'éditeur, cela pourrait permettre un meilleur confort de travail. Toujours dans cette même optique de confort, la gestion de l'historique de navigation pourrait être un atout pour l'application.

Il serait également envisageable d'ajouter un objet « option » à passer en paramètre de la librairie. La mise en place d'un site web de démonstration de l'application serait également un avantage non négligeable.

## Expériences et difficultés

### Difficultés rencontrées

Au cours de ce projet, nous avons dû surmonter certaines difficultés, qu'elles soient d'ordre organisationnel, technique ou humain. Voici une courte section expliquant certaines d'entre elles.

- **Hétérogénéité des compétences :** imputable à la formule proposée par l'année spéciale, nous avons des niveaux de compétences disparates, ce qui a engendré un certain manque de motivation de la part des acteurs du projet ayant une formation moins avancée. A cela s'est ajouté le manque de temps, ce qui a finalement abouti à une répartition inégale du travail.
- **Complexité du système :** la notion d'automate, bien qu'abordable, ne nous a pas été familière immédiatement, entraînant parfois des développements dénués de certaines fonctionnalités importantes pour un initié. Le recours à l'assistance de Monsieur Champin a été nécessaire à plusieurs reprises durant les réunions de ce point de vue.
- **Gestions des dépendances et modules :** La gestion des dépendances JavaScript de l'application a été compliquée à mettre en œuvre. En effet, il a été nécessaire à un certain point d'utiliser NPM dans ce but, ce qui a ajouté un frein supplémentaire à la participation pour les acteurs du projet n'ayant aucune notion dans la gestion de paquets propre à NodeJS. De plus, il a fallu générer une fonction unique à partir de nombreux modules, gérés initialement avec RequireJS. Nous avons rencontré de grandes difficultés

# Rapport de conception

à réunir tous ces modules avant d'adopter Webpack, un gestionnaire de module très simple d'accès qui nous a finalement permis d'arriver à nos fins.

## Changements depuis la première présentation

Le déroulement du projet a subi plusieurs changements depuis la mi-parcours. L'application présentée à cette première étape de rendu proposait un système de menu pour ouvrir l'application en mode « vue » ou « édition » qui a été supprimé à la faveur d'un paramètre à passer à la fonction « `fsa_editor` ». Le mode « création » initialement envisagé n'a finalement été qu'une extension du mode « édition », lui-même basé sur le mode « vue ». Nous avons incrémenté pas à pas les différents paramètres éditables grâce à l'interface proposée, jusqu'à arriver à l'état actuel pas encore exhaustif mais suffisant à l'édition basique d'un automate. L'ajout de la gestion de dépendances locales grâce à NodeJS et à son gestionnaire de paquets NPM a également représenté une modification considérable dans notre manière d'envisager le projet. Enfin, les commentaires du code, à l'origine purement indicatifs, ont été revus afin de correspondre aux normes JSDoc, permettant ainsi une génération automatique de la documentation du projet.

## Conclusion

### Perspectives pour l'éditeur graphique

Etant donnée la limite de temps qui délimitait le projet, il reste des améliorations à apporter à ce dernier de manière à le rendre davantage efficace.

Afin de pouvoir le rendre utilisable aux chercheurs travaillant sur ce projet, l'éditeur graphique sera intégré au site Web de Monsieur Champin, de manière à ce que l'objet puisse être modifié directement par l'intermédiaire de l'éditeur.

Nous espérons pouvoir, sur notre temps libre, continuer à contribuer au projet de manière à remplir, à terme, les fonctionnalités exigées par les utilisateurs en profitant de leurs retours éventuels.

### Apports personnels

Ce projet fut un réel apport, non seulement sur le plan personnel mais aussi sur le plan collectif. En effet, ce fut pour tout le monde l'occasion de se familiariser avec les Editeur graphique d'automate fini

# Rapport de conception

démarches à suivre en termes de gestion de projet de manière à répondre aux attentes de Monsieur Champin. Egalement, ce fut l'occasion pour certains de découvrir les technologies Web telles que JavaScript, HTML ou encore CSS.

Ce fut d'autant plus intéressant que, étant issu de l'année spéciale, nous avons tous un parcours et des compétences différentes que ce soit dans la gestion de projet, le domaine technique ou encore la rédaction de rapports, et nous avons tous eu la chance d'apprendre des autres ce qui fut particulièrement enrichissant.

Il est intéressant de noter que Martin Bolot, l'un des acteurs du projet, a pu bénéficier de la présentation de ce projet tuteuré pour l'obtention d'un stage dans le domaine de l'édition graphique en JavaScript.

# Rapport de conception

## Annexes

### Webographie

<https://developer.mozilla.org/>

<http://stackoverflow.com/>

<https://openclassrooms.com/>

<http://liris.cnrs.fr/~pchampin/enseignement/intro-web/>

<http://liris.cnrs.fr/~pchampin/enseignement/intro-js/>

<http://www.alsacreations.com>

### Dépôt GitHub

Adresse du dépôt GitHub contenant le code source de l'application :

<https://github.com/EditeurGraphiqueAutomateFini/FSA-Editor>

### Table des illustrations

Figure 1 - Planning.....	4
Figure 2 - Bête à cornes.....	5
Figure 3 - Diagramme FAST .....	6
Figure 4 - Schéma de l'applicaion.....	9



# Rapport de conception

## Remerciements

Dans le cadre de ce projet, nous tenions à remercier les différents acteurs qui ont contribué à celui-ci, de près ou de loin :

- **Pierre-Antoine Champin**, notre tuteur. Toujours disponible et enthousiaste, il nous a guidé tout au long de cette aventure.
- **Vincent Vidal**, candide lors de notre soutenance. Des conseils avisés et des questions permettant de mieux cerner certaines problématiques.
- **Amélie Cordier**, responsable des projets tuteurés.
- **Jocelyne Debouté**, responsable de l'année spéciale.
- **Anne Corrigan**, directrice du département informatique.

Toute l'équipe projet est fière d'avoir contribué, à son niveau, aux différents travaux de Monsieur Champin. Tout n'a pas toujours été simple, Martin Bolot membre de l'équipe projet a beaucoup apporté à ce projet grâce à sa passion pour le JavaScript et à son expérience dans les technologies Web.