



MOTEUR DE JEU DE STRATÉGIE WEB

PROJET TI5

Veille technologique

Auteurs :

Ilyas BOUTEBAL

Maxime COLIN

Adrian GAUDEBERT

Youness HAMRI

Van Duc NGUYEN

Client :

Pierre-Antoine CHAMPIN

20 janvier 2011

Table des matières

1	Compatibilité	3
1.1	Propriétés CSS3	3
1.2	Applications Web HTML5	4
1.3	Graphisme et contenu imbriqué HTML5	4
2	Client	5
2.1	HTML5	5
2.2	Canvas	5
2.2.1	Points forts	5
2.2.2	Points faibles	5
2.3	SVG	6
2.3.1	Points forts	6
2.3.2	Points faibles	6
2.4	Flash	6
2.5	Crafty.js	6
2.5.1	Points forts	7
2.5.2	Points faibles	7
2.6	Comparatif	8
2.7	Conclusion	8
3	Serveur	9
3.1	Twisted	9
3.2	Tornado	10
3.3	Comet	10
3.4	AJAX Push Engine Project	10
3.5	Ruby Event Machine	11
3.6	Node.js	12
3.7	Conclusion	12
4	Communication	14
4.1	WebSocket	14
4.1.1	Présentation	14
4.1.2	Implémentation et support coté client	14
4.1.3	Implémentation et support coté serveur	14
4.1.4	Sécurité	15
4.2	Ajax	15
4.2.1	Présentation	15
4.2.2	Support	15
4.3	Opera Unite	15
4.3.1	Présentation	15
4.3.2	Implémentations	15
4.3.3	Conclusion sur Opera Unite	15
4.4	Conclusion	16

5	Formats de données	17
5.1	JSON	17
5.1.1	Présentation	17
5.1.2	Exemple	17
5.1.3	Sources	17
5.2	XML	18
5.2.1	Présentation	18
5.2.2	Exemple	18
5.2.3	Sources	18
5.3	HTML	18
5.3.1	Présentation	18
5.3.2	Sources	18
5.4	Conclusion	18
6	Choix technologiques	19

Chapitre 1

Compatibilité

Le HTML5 et CSS3 sont des nouveautés à suivre ces temps-ci, et les navigateurs commencent à s'y adapter, mais à quel point ?

Voici quelques sites qui permettent de tester son navigateur, et de faire un petit bilan sur ce qui fonctionne, sur quel navigateur etc. . .










Html5 test.com : Ce site permet de réaliser un test de comptabilité d'un navigateur avec html5, et d'obtenir le résultat sous la forme d'un score sur 300.

caniuse.com : Dans ce site on trouve des tableaux récapitulatifs en fonction des différents standards et des navigateurs. Par contre, ce n'est pas très facile de s'y retrouver, mais le rappel des anciennes versions permet d'avoir une vue global de l'évolution.










FindMeByIP.com : C'est une application Web gratuite qui permet de déterminer quelles sont les caractéristiques valides des différents navigateurs, en particulier concernant : HTML5 et CSS3. L'outil analyse et fournit les informations concernant la compatibilité de chaque navigateur.

Pour conclure, quelques tableaux qui montrent la compatibilité de plusieurs navigateurs sur différents OS avec les différentes technologies.







1.1 Propriétés CSS3

OS	MAC				WIN							
Navigateur	 OPERA	 FIREFOX	 SAFARI	 CHROME	 OPERA	 FIREFOX	 SAFARI	 IE	 CHROME			
Version	10.63	3.6	5	7	10.63	3.6	4.03	5	6	7	8	9
Multiple Backgrounds	O	O	O	O	O	O	O	O	N	N	N	O
Opacity	O	O	O	O	O	O	O	O	N	N	N	O
CSS Animations	N	N	O	O	N	N	N	O	N	N	N	O
CSS Transforms 3D	N	N	O	N	N	N	N	O	N	N	N	N
CSS Transitions	O	N	O	O	O	N	O	O	N	N	N	O

1.2 Applications Web HTML5

Platform	MAC				WIN							
Navigateur	 OPERA	 FIREFOX	 SAFARI	 CHROME	 OPERA	 FIREFOX		 SAFARI	 IE		 CHROME	
Version	10.63	3.6	5	7	10.63	3.6	4.03	5	6	7	8	9
Local Storage	O	O	O	O	O	O	O	O	N	N	O	O
Session Storage	O	O	O	O	O	O	O	O	N	N	O	O
Offline Applications	O	O	O	O	O	O	O	O	N	N	N	N
Drag and Drop	N	O	O	O	N	O	O	O	O	O	O	O
Hash Change (Event)	O	O	O	O	O	O	O	O	N	N	O	O
History Management	N	N	O	O	N	N	O	O	N	N	N	N
Web Sockets	N	N	O	O	N	N	O	O	N	N	N	N

1.3 Graphisme et contenu imbriqué HTML5

Platform	MAC				WIN							
Navigateur	 OPERA	 FIREFOX	 SAFARI	 CHROME	 OPERA	 FIREFOX		 SAFARI	 IE		 CHROME	
Version	10.63	3.6	5	7	10.63	3.6	4.03	5	6	7	8	9
Canvas	O	O	O	O	O	O	O	O	N	N	N	O
Canvas Text	O	O	O	O	O	O	O	O	N	N	N	O
SVG	O	O	O	O	O	O	O	O	N	N	N	O
SVG Inline	N	N	N	O	N	N	O	N	N	N	N	O
Audio	O	O	O	O	O	O	O	O	N	N	N	O
Video	O	O	O	O	O	O	O	O	N	N	N	O

Donc la question qui se pose est : faut-il utiliser ces nouvelles technos (HTML5, CSS3, canvas...) dès aujourd'hui? Nous pensons que oui parce que :

- Les balises HTML5 sont déjà en partie supportées par les navigateurs récents : Firefox, Safari, Chrome...
- Le HTML5 Canvas permet de créer des applications dynamiques / animées / de qualité équivalente à Flash... sans Flash!
- La lecture native de contenu media, vidéo et audio, dans la plupart des navigateurs.
- On peut utiliser « Chrome Frame » pour rendre compatible l'HTML5 sur les vieux navigateurs comme IE6, IE7.
- Les propriétés CSS3 sont déjà en partie supportées par plusieurs navigateurs récents comme Firefox, Safari, Chrome...

Chapitre 2

Client

2.1 HTML5

HTML5 est la dernière version de HTML, il introduit dans ses spécifications plusieurs balises, de nouveaux types et attributs pour les anciennes balises. On pourrait implémenter la majorité des fonctionnalités qu'on souhaite en HTML, couplé avec CSS et JavaScript, mais ce langage n'est pas conçu pour nos besoins, et on va donc chercher d'autres solutions plus adaptées.

2.2 Canvas

Canvas est un composant de HTML5 qui permet d'effectuer des rendus dynamiques d'images bitmap via des scripts. Canvas se résume donc en une zone de dessin dont la hauteur et la largeur sont définies dans du code HTML. Du code javascript permet d'accéder à l'aire via une série complète de fonctions de dessins, similaires aux autres API 2D, bien que permettant de générer dynamiquement des graphismes. Certaines personnes ont anticipé cet emploi de canvas en l'utilisant pour des graphiques, des animations et de la création d'images. Le composant canvas est plutôt récent et n'est pour le moment implémentée que par quelques navigateurs : Firefox, Opéra, Safari, Konqueror, Google Chrome. Internet Explorer est encore en retard là-dessus mais l'arrivée de IE 9 permettra de le rattraper.

2.2.1 Points forts

- Dans un contexte hautement dynamique, la légèreté des images bitmap permet un traitement plus rapide des graphiques.

2.2.2 Points faibles

- La qualité des graphiques baisse en fonction de la taille de l'écran.
- Très jeune âge.

Sources :

- [http://fr.wikipedia.org/wiki/Canvas_\(HTML\)](http://fr.wikipedia.org/wiki/Canvas_(HTML))
- <https://developer.mozilla.org/fr/HTML/Canvas>

2.3 SVG

SVG est un format de données conçu pour décrire des ensembles de graphiques vectoriels. Il est basé sur XML. Les coordonnées, dimensions et structures de ces graphiques vectoriels sont indiqués sous forme numérique dans le document XML. Les couleurs et les polices de caractères à utiliser sont gérées par les feuilles de styles. En plus de la gestion de formes basiques (rectangles, ellipses, etc.), SVG gère des chemins (paths), qui utilisent les courbes de Bézier et qui permettent ainsi d'obtenir presque n'importe quelle forme. D'autres outils lui permettent de gérer également le remplissage et la transparence des graphiques.

2.3.1 Points forts

- Les graphiques vectoriels permettent de s'adapter aux différentes tailles d'écran, i.e. graphiques de très bonne qualité en toute circonstance.
- Disponibilité de bibliothèques Javascript qui proposent des fonctionnalités de base et permettent de faciliter le travail des développeurs (i.e. raphael.js).
- Il existe des logiciels graphiques qui permettent de modifier facilement chaque forme, par exemple en déplaçant des points, ou en changeant la couleur des traits, etc.

2.3.2 Points faibles

- Le revers de la médaille de la qualité du graphisme est évidemment la consommation accrue de CPU (proportionnelle à la taille du graphique),
- Rendement très faible dans un contexte très dynamique.
- Ne permet pas de créer des points d'articulations, tels des noeuds dans un graphe. Bref, la notion de pointeur n'existe pas en SVG, ce qui rend la description de scènes dynamiques complexe.

Sources : http://fr.wikipedia.org/wiki/Scalable_Vector_Graphics

2.4 Flash

Les fichiers Flash, généralement appelés « animation Flash » portent l'extension .swf. Ils peuvent être inclus dans une page web et lus par le plugin Flash du navigateur, ou bien interprétés indépendamment dans le lecteur Flash Player. Flash est incontestablement une des méthodes les plus populaires pour ajouter des animations et des objets interactifs à une page web. Cependant, si Flash est une technologie très adaptée à nos besoins, elle est propriétaire et n'est donc pas compatible avec notre volonté de faire du « Web Ouvert ».

Sources :

- <http://pro.clubic.com/creation-de-site-web/langage-programmation/actualite-334796-cs5-flash-car.html>
- http://fr.wikipedia.org/wiki/Adobe_Flash#Flash_et_le_.C2.AB_Web_.C2.BB

2.5 Crafty.js

Crafty.js est un framework JavaScript pour les jeux vidéos, dont le but est de fournir un ensemble de briques de bases et d'être très modulable. Il se base pour cela sur le modèle Composant / Entité¹ : chaque objet du jeu est une entité, et chaque entité est composée d'un ensemble de composants. Les

1. Cf. <http://cowboyprogramming.com/2007/01/05/evolve-your-heirarchy> ou <http://www.gamearchitect.net/Articles/GameObjects1.html>

différents composants de l'application contiennent les fonctionnalités du jeu. On aura par exemple une entité Unité qui aura pour composants “2D, draw, health, clickable”.

Utiliser Crafty dans ce projet apporte plusieurs avantages : tout d'abord, on bénéficie d'une structure de base pour le code côté client, Crafty.js étant un framework qui fournit une structure et des fonctionnalités utiles à notre projet. Le modèle Composant / Entité présenté ci-dessus présente de nombreux intérêts, et nous permettra d'étendre de manière simple notre code, en créant de nouveaux composants. Ensuite, Crafty nous permettra, dans un premier temps au moins, de ne pas nous occuper de fonctionnalités comme l'affichage, afin de nous concentrer sur des éléments plus importants (le gameplay par exemple).

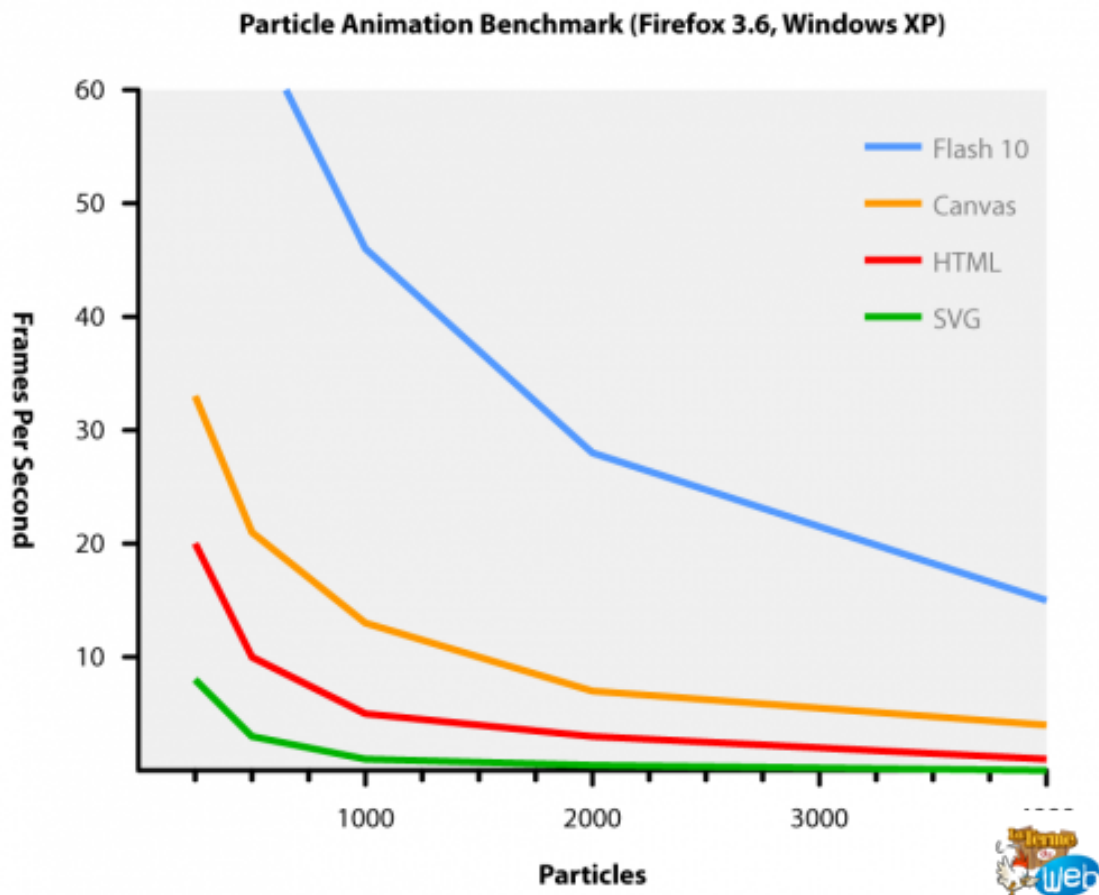
2.5.1 Points forts

- Fourni un grand nombre de fonctionnalités nécessaires à notre moteur
- Architecture modulaire et extensible (possibilité d'ajouter, de manière simple, nos propres fonctionnalités)
- Licence libre (double licence GPL et MIT)

2.5.2 Points faibles

- Pas de prise en charge native de SVG (uniquement canvas et DOM), mais possibilité d'étendre
- Framework très récent (actuellement en version 0.2) et maintenu par un seul développeur

2.6 Comparatif



Le graphique ci-dessous nous donne un comparatif des quatre technologies dans un contexte très dynamique.

Le graphique ci-dessus donne un comparatif du rendement de ces quatre technologies sous Firefox (résultat similaire pour les autres navigateurs) et le résultat est sans appel : Flash l'emporte haut la main, canvas vient en deuxième position, HTML et SVG en dernière position. Le lien suivant donne un comparatif en image du rendement de ces technologies : <http://www.lafermeduweb.net/billet/un-bench-flash-vs-canvas-comparant-une-animation-de-particules-801.html>

2.7 Conclusion

Malgré la puissance de Flash, nous avons décidé de ne pas utiliser cette technologie pour rester dans l'esprit de l'open source. Dans le contexte de notre moteur de jeux qui n'est pas très dynamique puisqu'au tour par tour, le large panel d'outils offert par SVG, grâce à son ancienneté et la qualité de ses graphiques, nous incite à choisir cette technologie. De plus, elle est largement supportée par les navigateurs (cf. Etude sur la compatibilité).

L'outil Crafty.js nous paraît également approprié afin de gagner du temps lors du développement. Nous choisissons donc de l'utiliser avec pour but d'y implémenter la gestion du SVG.

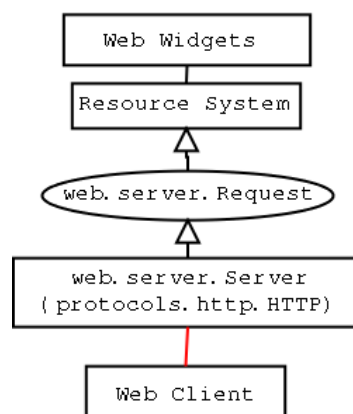
Chapitre 3

Serveur

3.1 Twisted

Twisted est un moteur réseau événementiel (event-driven networking engine) écrit en Python et sous licence MIT Free Software licence. Il contient un serveur Web, de nombreux client chat, de serveur chat, de serveur mail, et etc. Twisted supporte TCP, UDP, SSL/TLS, multicast, Unix domain sockets, un grand nombre de protocoles dont HTTP, NNTP, IMAP, SSH, IRC, FTP, et beaucoup d'autres.

- Le code source complet contient de nombreux Hooks pour le contenu dynamique.
- Écrit en Python, un langage de haut niveau, Twisted offre la sécurité et la stabilité grâce à la classe la plus commune de réseau sécurité, le "buffer-overflow", et au mécanisme de Python pour gérer des erreurs.
- Servant les sites web à la haute circulation, le serveur web Twisted.web est configuré facilement.
- Plusieurs APIs intégrés permettent aux développeurs de développer rapidement des nouveaux protocoles et des nouveaux services.
- Simplement pour développer le client et serveur avec une même base de code (base-code).



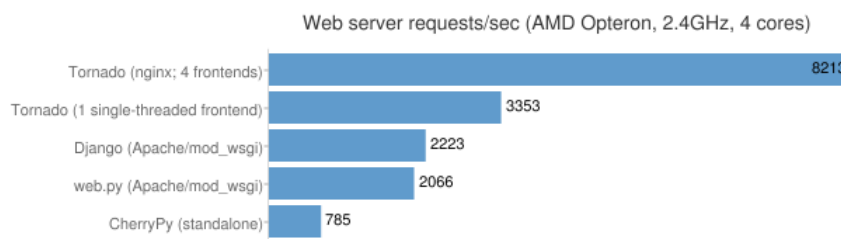
Lorsque le Serveur reçoit une requête de Client, il crée un objet Request et l'envoie au système de ressources qui détermine l'objet Resource approprié à l'URL demandé par le Client.

3.2 Tornado

Tornado est la version open source de serveur web de FriendFeed racheté par Facebook. C'est un serveur d'application web asynchrone et non-blocking, écrit en Python. Il peut traiter de nombreuses connexions simultanée (les services Web en temps réel).

Tornado contient plusieurs modules : web, escape, database, template, httpclient, auth, locale, option, ... Le module le plus important est le module web qui inclut la plupart des fonctionnalités principales de Tornado. Les autres modules sont les outils pour assister ce module.

Parmi les frameworks web Python, Tornado est le plus rapide.



3.3 Comet

Ce sont les techniques permettant à un serveur web d'envoyer des informations au navigateur web sans que celui-ci l'ait explicitement demandé. Il dépend des composants inclus par défaut dans les navigateurs web, tel que JavaScript. Cette approche diffère amplement du modèle classique du web où le navigateur demande un page web complet pour chaque fois.

Des méthodes spécifiques implémentant Comet tombe dans 2 major catégories :

- Streaming : Hidden IFrame, XMLHttpRequest
- AJAX avec long polling : XMLHttpRequest long polling, Script tag long polling.

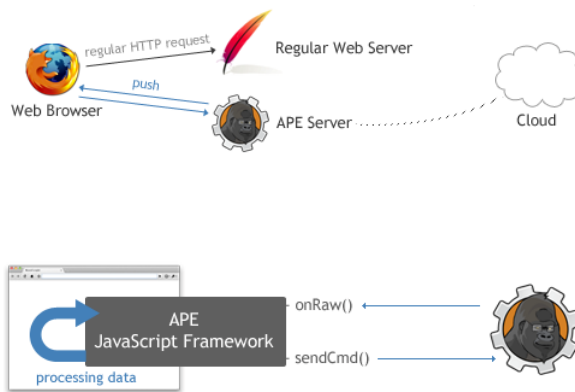
3.4 AJAX Push Engine Project

APE est une technologie permettant d'échanger des données entre des milliers d'utilisateurs via un navigateur web, sans recharger tout la page web.. APE n'utilise que les standards du web (avec AJAX), alors il est compatible avec tous les navigateurs web sur les ordinateurs et bien sur les appareils mobiles.

APE est divisé en 2 parties distinctes : La partie la plus centrale est le serveur APE étant le serveur epoll-driven HTTP streaming. Et la partie APE JavaScript Framework envoie et reçoit les actions du côté client grâce aux les protocoles APE.

- **Serveur APE** : étant un serveur COMET et écrit en C, le serveur APE implémente les méthodes POST et GET du protocole HTTP. Il est utilisé uniquement pour AJAX Push.
- **APE JavaScript Framework** : reçoit des informations envoyées par le serveur (RAW), gère les données, et renvoie les commandes d'utilisateurs (CMD).

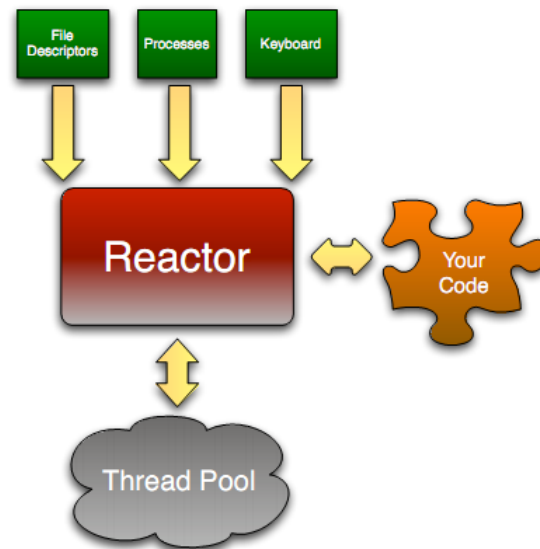
L'avantage de l'APE le plus important est la vitesse du chargement. Il est entièrement adapté pour les connexions lentes telles que EDGE ou 3G.



3.5 Ruby Event Machine

Eventmachine est une bibliothèque de programmation pour Ruby, C++, et Java. Il fournit event-driven I/O en utilisant le reactor pattern. Le reactor pattern décrit un gestionnaire de service qui reçoit les événements et les envoie pour enregistrement. L'avantage du reactor pattern est la séparation entre répartition d'événements et l'application logique qui gère les événements, sans compliquer le code avec le multithreading.

Fondamentalement, eventmachine (EM) s'occupe de toutes ces choses au bas niveau : l'écoute sur les sockets, la création de connexions réseau, la gestion de TIMERS et des simultanités.



Eventmachine est conçu pour satisfaire simultanément 2 besoins principaux :

- L'expansion, les performances et la stabilité
- Une API qui élimine la complexité de la haute performance de programmation réseau, permettant

aux ingénieurs de se concentrer sur leur logique d'application.

L'idée de base : Au lieu d'attendre une réponse du réseau, EventMachine utilise ce temps pour traiter d'autres demandes. Il peut gérer 5000-10000 connexions simultanées avec un seul processus Ruby.

3.6 Node.js

Node.js est un framework Javascript qui s'exécute côté serveur, il est construit sur le moteur V8 (compilation à la volée), le très performant moteur JavaScript open source soutenu par Google et utilisé par le navigateur Chrome. Il permet de réaliser des applications web non-bloquantes, temps-réel (asynchrone), performantes, scalables.

Node.js utilise une boucle d'événements (event loop) au lieu de threads, et peut gérer des millions de connexions simultanées. Chaque opération I/O dans Node.js est asynchrone, ce qui signifie que le serveur peut continuer à traiter les demandes entrantes pendant que l'opération I/O se déroule.

Node.js fonctionne sur Linux, Mac OS X, FreeBSD. Windows ne supporte pas encore Node.js complètement.

Les avantages du langage JavaScript :

- Le nombre de programmeur JavaScript est géant et il continue à augmenter.
- JavaScript est soutenu par presque tous les navigateurs (même sur les appareils mobiles).
- Il est en train de devenir l'une des langues les plus populaires.

Les avantages de Node.js :

- JavaScript est fait pour l'évènementiel : la programmation par callback est familière aux développeurs d'applications AJAX et, pour se faire, la syntaxe des fonctions anonymes et le support des closures est adapté.
- Node est construit sur Javascript : de part sa présence dans les navigateurs, c'est peut-être le langage le plus programmé au monde et bénéficie de 15 ans d'expérience.
- Node se stabilise : l'API de Node gagne en maturité et est proche de se finaliser.
- Node est simple et petit : la documentation se promène d'un seul regard et permet de rapidement connaître ses fonctionnalités.
- Node est rapide : le moteur V8 et l'architecture non bloquante en font l'une des architectures les plus puissantes du marché, spécialement pour les requêtes longues et intensives en I/O.

3.7 Conclusion

En fait, tous ces technologies de serveur ont des avantages qui permettent de réaliser des serveurs web non-bloquant, asynchrones, performants et scalables.

En premier lieu, on sait qu'on ne voudra pas utiliser Ruby Event Machine, car il est écrit dans des langages dont nous n'avons pas les compétences. On préférera donc utiliser soit Python, soit JavaScript.

APE Project pourrait être un choix judicieux si on choisit également de ne pas utiliser les WebSockets : il fournit une alternative élégante et fonctionnelle.

Node.js a l'avantage d'utiliser le langage JavaScript, qui est également le langage côté client. L'utiliser permettrait donc de ne maîtriser qu'un seul langage, et donc de gagner en efficacité sur le projet. De plus, il semble très bien adapté à nos besoins.

Les serveurs Twisted et Tornado se distinguent peu au niveau des fonctionnalités dont nous avons besoin pour ce projet.

Sources

- <http://twistedmatrix.com>

- <http://www.tornadoweb.org/documentation>
- [http://en.wikipedia.org/wiki/Comet_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))
- <http://www.ape-project.org/ajax-push.html>
- <http://nodejs.org/>

Chapitre 4

Communication

Cette partie concerne les technologies de communication client/serveur et client/client. Les différentes pistes retenues sont les WebSockets, Ajax, Opera Unite.

4.1 WebSocket

4.1.1 Présentation

WebSocket est une technologie fournissant un canal de communication bidirectionnel et full-duplex à travers un socket TCP. Il a été conçu pour être implémenté dans les navigateurs et serveurs web, mais il peut être utilisé dans n'importe quelle application client ou serveur. L'API WebSocket est en phase de standardisation par le W3C et le protocole par IETF.

Un tel canal de communication permet :

- la notification au client d'une modification d'état du serveur
- l'envoi de données du serveur au client sans que celui-ci n'ait à faire de requête.

4.1.2 Implémentation et support côté client

Le protocole WebSocket est implémenté dans les navigateurs Chrome 4, Safari 5, Firefox 4 et Opera 11. Son support est néanmoins désactivé dans Firefox 4 et Opera 11 pour des raisons de sécurité. Il est possible de l'activer via dans les paramètres des deux navigateurs. Internet Explorer ne supporte pas WebSocket.

4.1.3 Implémentation et support côté serveur

Le protocole nécessite également d'être implémenté côté serveur pour être utilisé. Il existe plusieurs implémentations côté serveur de WebSocket, dans différents langages (Java, Python, PHP, Javascript...) et sous différentes formes (extension apache, serveur entier, script...).

Quelques implémentations :

- GNU WebSocket4J, une implémentation du protocole WebSocket en Java.
- pywebsocket3, une implémentation en Python sous la forme d'une extension pour le serveur Apache.
- jWebSocket, implémentation Java côté serveur et JavaScript/HTML5 côté client.
- Implémentation de WebSocket avec node.js

4.1.4 Sécurité

WebSocket comporte à l'heure actuel une faille de sécurité dans la phase de "handshake" permettant de remplacer un fichier javascript par un malware. La faille se situe au niveau de l'API elle-même. C'est pourquoi son support est désactivé par défaut sous Firefox 4 et Opera 11 jusqu'à ce que la faille soit fixée.

4.2 Ajax

4.2.1 Présentation

Ajax (Asynchronous Javascript and XML) est un rassemblement de différents outils et méthodes de conception permettant de construire des applications web dynamiques basées sur différentes technologies web côté client.

Ajax est une combinaison de technologies telles que JavaScript, XML, DOM et XMLHttpRequest dans le but de réaliser des applications Web qui offrent une maniabilité et un confort d'utilisation supérieur à ce qui se faisait jusqu'alors.

DOM et JavaScript sont utilisés pour modifier l'information présentée dans le navigateur par programmation. L'objet XMLHttpRequest est utilisé pour dialoguer de manière asynchrone avec le serveur Web. La notation XML est utilisée pour structurer les informations transmises entre le serveur Web et le navigateur.

En alternative au format XML, les applications Ajax peuvent utiliser les fichiers texte ou JSON.

4.2.2 Support

Les applications Ajax fonctionnent sur tous les navigateurs Web qui mettent en oeuvre les technologies décrites précédemment, parmi lesquels Mozilla Firefox, Internet Explorer, Konqueror, Google Chrome, Safari et Opera.

4.3 Opera Unite

4.3.1 Présentation

Opera Unite est une technologie qui transforme un navigateur en serveur Web personnel. Avec Opera Unite, on devient à la fois client et serveur, à la fois visiteur et hôte. On reçoit du contenu du Web, et on en fournit également. On garde le contrôle : les données restent sur votre ordinateur, et on décide avec qui on désire les partager.

4.3.2 Implémentations

Opera Unite est implémenté dans le navigateur Opera depuis la version 10.

4.3.3 Conclusion sur Opera Unite

Opera Unite est au final un service de partage de contenu et non de communication client/serveur, cette solution est donc inadaptée à nos besoins. De plus, cette fonctionnalité n'est disponible que sur le navigateur Opera.

4.4 Conclusion

En conclusion, la technologie WebSocket semble la plus intéressante pour notre projet. Elle semble adaptée à nos besoins. Néanmoins le fait que le protocole soit encore en phase de développement et le fait qu'il soit désactivé par défaut sur certains navigateurs pourrait poser problème.

Ajax semble également correspondre à certaines de nos attentes, mais dispose de plus de restrictions au niveau communication client/serveur.

Chapitre 5

Formats de données

5.1 JSON

5.1.1 Présentation

JSON (JavaScript Object Notation) est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l'information structurée. Créé par Douglas Crockford, il est décrit par la RFC 4627 de l'IETF.

Un document JSON ne comprend que deux éléments structurels :

- des ensembles de paires nom / valeur ;
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- des objets ;
- des tableaux ;
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.

Le format JSON est très facilement exploitable et manipulable en Javascript. Un document JSON représente un objet. Il est donc potentiellement plus facile à manipuler qu'un document XML.

5.1.2 Exemple

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

5.1.3 Sources

- <http://fr.wikipedia.org/wiki/Json>

5.2 XML

5.2.1 Présentation

Extensible Markup Language est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents. Ce langage est qualifié d'extensible car il permet à l'utilisateur de définir les balises des éléments. L'utilisateur peut multiplier les espaces de nommage des balises et emprunter les définitions d'autres utilisateurs

5.2.2 Exemple

```
<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>
```

5.2.3 Sources

- <http://fr.wikipedia.org/wiki/XML>

5.3 HTML

5.3.1 Présentation

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

5.3.2 Sources

- <http://fr.wikipedia.org/wiki/JSON>

5.4 Conclusion

JSON et XML sont les deux formats qui semblent les plus adaptés à nos besoins. JSON a un avantage au niveau de son interprétation par Javascript, une technologie certainement clé dans ce projet.

Chapitre 6

Choix technologiques

Au terme de cette étude, nous avons fait les choix suivants :

- Client : SVG + Crafty.js
- Serveur : Node.js
- Communication : WebSocket + JSON

Ces choix impliquent que l'on programmera exclusivement en JavaScript (avec Node.js côté serveur), avec bien sur du HTML et CSS côté client. De plus, nous utiliserons certainement certains outils (modules, bibliothèques...) répondant à nos besoins, si ceux-ci existent déjà, sont éprouvés et peuvent être isolés dans notre application, afin de rester globalement indépendant à ces outils.