

14 Cartographie Web et mobile

Michael P. Peterson, Etats-Unis

Traduction : Clément Poncet et Frank Fuchs

Harmonisation : Felix de Montety

Avant-propos

La cartographie a toujours été dépendante d'un ensemble d'outils, que ce soit pour mesurer le monde ou pour fabriquer la carte. Depuis 2005, un nouveau panel d'outils en ligne ou dans le cloud sous forme d'API (Interfaces de Programmation d'Applications) a été développé pour réaliser les cartes. Ces outils ont pour intérêt d'être accessibles automatiquement à toute personne possédant une connexion Internet. Ce chapitre explore cette nouvelle ère de la cartographie et présente la manière de créer et distribuer des cartes en utilisant des outils de cartographie gratuits.

14.1 Introduction

Il est difficile d'accroître l'importance des cartes en tant que formes de communication sur le monde. Elles nous aident à comprendre à la fois ce qui nous entoure et l'espace situé hors de notre perception directe. Les cartes ont une influence sur notre vision du monde et sur notre comportement. Elles nous connectent à notre environnement. Chacun d'entre nous est un cartographe puisque nous créons tous des cartes mentalement. Parfois, nous avons même besoin de dessiner ces cartes à d'autres pour expliquer comment trouver un endroit particulier.

La création de cartes et l'analyse des informations sous-jacentes ont évolué vers une science et sont de précieuses compétences pour de nombreux types de travaux. La plupart des outils de cartographie sont à présent disponibles dans le cloud, un système sophistiqué mêlant matériel informatique et logiciels accessibles par Internet. Les outils de cartographie cloud



Order Free Web Hosting

I want to host my own domain (domain must be registered already)

www.

or, I will choose your free subdomain (recommended)

www. .hostei.com

Your name

Your email (account details will be sent there)

Password (at least 6 symbols, both letters and numbers)

Type password again

Figure 14.1a. Page d'inscription du site <http://000webhost.com>. Les informations sur le compte sont affichées sur la droite. Il est possible de choisir un sous-domaine gratuitement. Un domaine spécifique avec un nom du type <http://www.peterson.com> vous sera facturé, mais choisir un nom de domaine comme <http://geographyproof.hostei.com> est gratuit (© 2014 First Class Web Hosting)

permettent des formes très avancées de cartographie. Un avantage évident de ces cartes en ligne est leur facilité de mise à disposition des autres.

Créer des cartes dans le cloud nécessite l'utilisation d'un serveur. Même s'il est possible de transformer presque tout ordinateur en serveur, il est plus facile et plus sécurisant de choisir un service d'hébergement en ligne.

Les services d'hébergement en cloud comme Amazon Web Services et Microsoft Azure intègrent l'évolutivité dans le sens où ils peuvent être mis à niveau pour servir des millions d'utilisateurs simultanés. Ces services sont facturés à l'espace disque et au cycle de traitement du serveur.

» Account Information	
Domain	geographyprof.hostei.com
Username	a8040697
Password	*****
Disk Usage	0.2 / 1500.0 MB
Bandwidth	100000 MB (100GB)
Home Root	/home/a8040697
Server Name	server33.000webhost.com
IP Address	64.120.177.162
Apache ver.	2.2.13 (Unix)
PHP version	5.2.*
MySQL ver.	5.0.81-community
Activated On	2011-03-10 16:23
Status	Active

Figure 14.1b. Informations sur le compte

Les hébergeurs qui ne facturent que l'espace disque (comme 000webhost.com et podserver.info.) constituent une autre option pour héberger des pages web. Même si ces services ne sont pas évolutifs, ils sont plus simples à utiliser et proposent des offres avec sous-domaine gratuit allant jusqu'à 1500 MB d'espace disque. Les sections suivantes décrivent comment créer un site Internet et mettre à disposition des cartes en utilisant un de ces services d'hébergement web gratuit.

14.2 Serveurs dans le cloud

14.2.1 Faire sa place dans le cloud

Les figures 14.1a et 14.1b présentent la page d'inscription de 000webhost.com et les informations sur le compte créé. Vous pouvez remarquer qu'un sous-domaine gratuit figure sous le nom de domaine hostei.com. Alors qu'une adresse comme <http://geographyprof.com> pourrait être plus pertinente, cela inclurait une facturation puisque cela représenterait un nouveau domaine. Afin d'obtenir un compte gratuit, il est important de ne pas enregistrer votre propre domaine.

Une adresse e-mail est nécessaire pour enregistrer le sous-domaine gratuit. La page d'information du compte explique que l'adresse web qui a été attribuée par l'hébergeur est la suivante <http://geographyprof.hostei.com> (ou <http://64.120.177.162>). Il est également précisé que 1500 MB d'espace disque sont disponibles. Les informations sur le compte présentent aussi la disponibilité du serveur web Apache, et d'autres outils en ligne tels que PHP et MySQL.

Une interface graphique est incluse dans la plupart des services proposés par les hébergeurs web. Cette interface est appelée panneau de contrôle, ou cPanel (voir fig. 14.2). Des outils gèrent les courriels, l'édition de fichiers, la planification des tâches et la gestion du compte. Tous ces outils sont des projets open-source qui sont écrits et entretenus par de petits groupes de développeurs. Le gestionnaire de fichiers est l'outil le plus utile pour gérer les fichiers et construire un site web. MySQL et phpMyAdmin sont utilisés pour construire une base de données. La plupart des hébergeurs web utilisent un panneau de contrôle similaire afin d'accéder aux ressources du serveur.

La figure 14.3 présente la fenêtre du gestionnaire de fichiers avec accès aux outils d'envoi vers le serveur ainsi que la création de nouveaux fichiers et répertoires (sous-dossiers). Il existe d'autres outils pour déplacer, supprimer et renommer les fichiers. La liste de fichiers présente le nom, le type, la taille du fichier, tandis que les champs "Owner", "Group", et "Perms" décrivent les paramètres de sécurité. "Mod Time" indique quand le fichier a été modifié pour la dernière fois. Les fichiers peuvent être édités directement depuis cette fenêtre en cliquant sur "Edit" à la fin de chaque nom de fichier.

Le dossier **public_html** est le répertoire dans lequel tous les fichiers web sont publiés. Si un fichier html doit être affiché sur une page web, il doit être inclus dans ce dossier. Habituellement, ce dossier contient un fichier nommé `index.htm` (ou `index.php`) qui est la première page à laquelle on a accès lorsque le site est utilisé. Par

exemple, si une adresse du type <http://geographyprof.hostei.com/CloudMapping/> est entrée dans un navigateur, ce dernier va chercher un fichier nommé `index.htm` dans un dossier nommé `CloudMapping` lui-même localisé dans le répertoire `public_html`. Cela signifie que les adresses suivantes afficheront le même fichier : http://geographyprof.hostei.com/Online_Mapping/ http://geographyprof.hostei.com/Online_Mapping/index.htm



Figure 14.2. Panneau de contrôle standard chez la plupart des hébergeurs web, appelé cPanel, qui donne accès à certains outils. Le gestionnaire de fichiers est le programme principal destiné à l'envoi et la modification de fichiers - (© 2014 First Class Web Hosting)



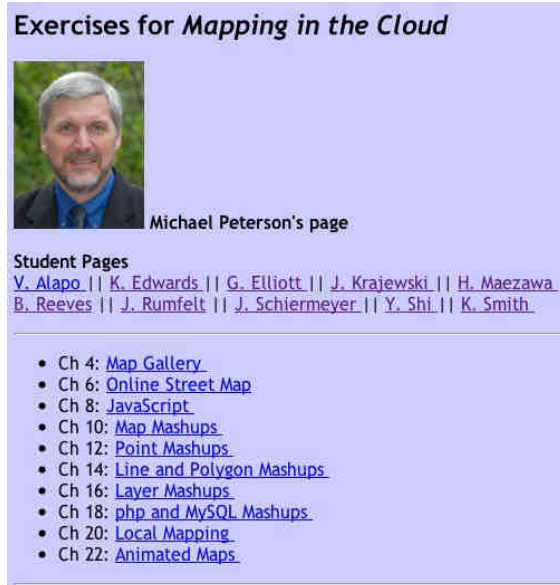
Figure 14.3. Fenêtre du gestionnaire de fichiers d'un hébergeur web. Ce service permet de créer et modifier des fichiers. Tous les fichiers destinés à un affichage sur le web doivent être placés dans le répertoire **public_html** (© 2014 First Class Web Hosting)

Normalement, le fichier `index.htm` sert de point d'entrée dans le site web et comporte tous les liens vers les autres fichiers du répertoire.

Le fichier `index.htm` aura une structure relativement simple - une ligne de titre suivie des liens de tous les fichiers associés. Ce fichier peut comporter une image du propriétaire du site et des liens vers les sites web d'autres étudiants, comme indiqué sur la figure 14.4. Le code permettant l'affichage d'une image est inséré en utilisant la balise

```
<img src=filename>
```

Les liens vers les pages des étudiants sont séparés par deux lignes verticales ("|"). Le code complet du fichier `index` peut être obtenu en cherchant "Peterson Mapping in the Cloud".



```
<html>
<head></head>
<body bgcolor="#CCCCFF">
<h2> Exercises for <i>Mapping in the
Cloud</i> </h2>
<img src=peterson.jpg height=150><b>
Michael Peterson's page
</b><p>
<p>
<b> Student Pages </b><br>
<a href=http://victoriaA.site88.net> V.
Alapo </a> ||
<a href=http://mapsarefuntoo.web44.net> K.
Edwards </a> ||
<br> <hr>
<ul>
<li>
Ch 4: <a
href=http://maps.unomaha.edu/onlinemapping/
code04.zip>
Map Gallery </a><br>
<li>
Ch 6: <a href=code06.zip> Online Street
Map</a><br>
```

```
</li>
Ch 8: <a
href=http://maps.unomaha.edu/onlinemapping/
code08.zip>
JavaScript </a><br>
</a><br>
</ul>
<hr>
</body>
</html>
```

Figure 14.4. Exemple de page index.htm incluant une image, des liens vers tous les autres étudiants, et des liens vers les pages associées. Le code montre comment afficher une image en utilisant la balise img, les liens vers les autres étudiants et vers les autres pages

14.2.2 HTML

Le HTML est la pierre angulaire du web. C'est le langage qui permet l'affichage d'information sur les pages web. C'est également un conteneur pour d'autres langages tels que JavaScript et PHP. Une certaine connaissance du HTML est nécessaire pour afficher des cartes sur Internet.

Le HTML est composé de balises qui définissent la mise en page (Willard, 2009). Il inclut de simples codes textuels, entourés par les séparateurs "<" et ">", qui spécifient comment le document va apparaître dans le navigateur. D'autres balises HTML créent également des liens vers des documents ou affichent des fichiers graphiques. Un éditeur de texte ordinaire peut être utilisé pour créer un fichier (Bloc-Notes sur Windows, ou TextEdit sur Macintosh, avec les paramètres appropriés). Contrairement aux logiciels de traitement de texte, ces programmes ont pour but de contenir du texte non-formaté. Une fois que les fichiers sont créés, ils peuvent être ouverts avec un navigateur tel Internet Explorer, Firefox ou Chrome.

Tous les fichiers HTML débutent par la balise "<html>" et finissent avec la même balise précédée d'un slash ("</html>"). Le slash précédant la balise finale indique que le code HTML est terminé. Techniquement, toutes

les balises HTML ont un début et une fin, cette dernière étant précisée par le "/" Par exemple, la balise <h1> est utilisée pour débiter une en-tête de texte - de taille plus grande pour faire des titres - et la balise </h1> termine le formatage de l'en-tête.

Nous pouvons à présent afficher une carte avec la balise **img**. Tous les noms de fichiers référencés avec **img** doivent se terminer par GIF, JPG (ou JPEG) ou PNG, car ce sont les extensions de fichiers les plus répandues dans les différents navigateurs. L'option "src" de la balise **img** est utilisée pour spécifier l'adresse URL du fichier. Sur l'exemple ci-dessous, remarquez que le type du fichier est le ".gif" présent à la fin de l'URL. La commande **img** inclut également un certain nombre d'options qui peuvent modifier la taille de l'image ou changer sa position dans la page. Comme hr, img n'a pas de balise de fin standard. Dans du formatage XHTML strict, il faut suivre ce code :

```
<img src=map.png />
```

La balise <embed> est utilisée pour afficher des fichiers graphiques qui ne sont pas au format GIF, JPEG, ou IMG. On peut prendre comme exemples les formats PDF, Flash, SVG, et QuickTime.

Le format est identique à celui de la balise :

```
<embed src="http://maps.unomaha
.edu/Cloud_Mapping/Chapter4/MapExample4.pdf"
width="500" height="389">
```

L'une des balises les plus utilisées en HTML est l'ancre, habituellement utilisée avec une chaîne de caractères pour créer un lien hypertexte. Elle peut également être utilisée avec la balise **img** pour créer un lien sur l'image. Le format de la commande d'ancrage est le suivant :

```
<a href=http://maps.unomaha
.edu/Cloud_Mapping/Chapter4/MapExample4.pdf>
Click for PDF file</a>
```

14.2.3 JavaScript

En soi, le HTML est simplement un langage de formatage de page. Combinée avec le JavaScript, une page HTML peut exécuter des codes informatiques (W3Schools.com 2011). D'abord réservée aux adeptes, la programmation est aujourd'hui vue comme une forme d'expression, comme un "développement de la pensée", et une compétence nécessaire. Ce concept "d'alphabétisation du codage" met en exergue de nouvelles idées concernant l'importance d'être capable de programmer. Des sites de programmation en ligne comme CodeAcademy sont devenus extrêmement populaires. La Khan Academy a introduit une suite similaire d'exercices de programmation en ligne. Les exemples suivants proposent une brève introduction à JavaScript et à la façon dont il peut être utilisé pour faire appel à d'autres fonctions.

CODE	Result
<pre><html> <body> <script type="text/javascript"> var x = 2 * 2 document.write("x = ", x) </script> </body> </html></pre>	<p>x=4</p>

Figure 14.5. Calcul réalisé avec JavaScript au sein du <body> d'un fichier HTML

Les fonctions sont les blocs fondamentaux du JavaScript. Une fonction est une procédure - une suite d'instructions qui exécutent une tâche spécifique. Les fonctions sont généralement définies dans la section <head> d'un document HTML. Cela permet de s'assurer que toutes les fonctions sont bien définies avant que tout contenu soit affiché. L'exemple dans la figure 14.6 définit une fonction simple dans la section <head> d'un document HTML. La fonction est par la suite appelée dans le <body> du document.

La fonction **square** prend un argument, appelé *number*, et elle n'a qu'une seule instruction : **return number * number**, qui indique qu'elle doit retourner (return) l'argument multiplié par lui-même. Cette instruction return spécifie la valeur qui est retournée comme résultat par la fonction.

Code	Result
<pre><head> <script LANGUAGE="JavaScript"> function square(number) { return number * number } </script> </head> <body> <script> document.write("The square of 5 is ", square(5), ".") </script> <P>All done.</P> </body></pre>	<p>The function returned 25. All done.</p>

Figure 14.6. Fonction mettant au carré le nombre indiqué lorsque la fonction est appelée

Le fichier externe qui contient les fonctions JavaScript peut se trouver sur le même ordinateur que le fichier HTML, comme présenté dans la figure 14.7, mais il peut également se trouver sur un autre ordinateur ou serveur. C'est ainsi que les codes API sont distribués. Faire référence à une bibliothèque d'un code API permet à un concepteur d'accéder à des milliers de fonctions.

Plutôt que d'enregistrer le code JavaScript directement dans le fichier HTML, soit dans le corps, soit dans l'entête, il est possible de placer les fonctions JavaScript dans un fichier séparé. L'attribut SRC du tag <script> spécifie où le code JavaScript peut être trouvé. La figure 14.8 montre le fichier externe appelé *common.js* et comment il est référencé dans la section d'entête d'un

document HTML. Le fichier externe JavaScript peut contenir de multiples fonctions contrairement au code HTML.

Code	Result
<pre>function square(number) { return number * number } <head> <title>Referencing a file of functions</title> <script src="common.js"> </script> </head> <body> <script> document.write("The square of 5 is ", square(5), ".") </script> <p>All done.</p> </body></pre>	<p>A separate file called "common.js"</p> <p>The function returned 25. All done.</p>

Figure 14.7. Fonction située dans un document externe, *common.js*. Cette fonction est ensuite référencée dans le fichier HTML avec <script src="common.js">

La figure 14.8 montre également le mode de référencement du code API Google Maps. Afin de faciliter le débogage, le code API Google Maps va fonctionner localement sur un ordinateur sans avoir besoin d'être transféré vers un serveur. Mais pour que d'autres puissent voir la carte, le code doit résider sur un serveur.

```
Code
<head>
<title>Google Maps JavaScript API
Example</title>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sen
sor=false">
</script>
</head>
```

Figure 14.8. Implémentation d'un appel vers l'API Google Maps qui accède à un nombre plus large de fonctions liées à la cartographie. Ici, le capteur est paramétré sur "false" (faux). La valeur du capteur serait paramétrée sur "true" (vrai) si un appareil mobile avec la fonction de localisation de position était utilisé

14.3 API Google Maps

Introduite peu après Google Maps en 2005, l'API Google Maps consiste en une série de fonctions qui contrôlent l'apparence de la carte, incluant l'échelle et l'emplacement, ainsi que toute autre information sous forme de points, lignes ou zones et descriptions associées. L'utilisation de l'API Google Maps est gratuite tant que le site qui l'utilise ne fait pas payer l'accès à ses visiteurs. Google fixe une limite sur le nombre de cartes qui peuvent être utilisées. Un site ne peut pas générer plus de 25 000 cartes par jour sur 90 jours consécutifs. Une carte générée équivaut à une carte affichée avec l'API Google Map. Une fois chargée, le niveau d'utilisation de la carte par l'utilisateur n'a pas d'impact sur le nombre de chargements de cartes. Il serait extrêmement difficile pour l'utilisateur moyen de l'API Google Maps de dépasser les 25 000 chargements de cartes. Même si un site devenait "viral", il devrait rester sur un rythme de plus de 25 000 cartes chargées par jour pendant 90 jours consécutifs avant que la limite ne soit dépassée. De plus, les limites d'utilisation peuvent figurer sur le site afin que la limite ne soit pas dépassée. Si un site dépasse sans arrêt les 25 000 cartes par jour, Google vous demandera d'inscrire votre site et de payer

0,5\$ (environ 0,35 €) par tranche de 1 000 cartes chargées au-delà de cette limite. Par exemple, si votre page de cartographie était sollicitée 100 000 fois par jour, 90 jours de suite, vous seriez obligé(e) de payer 37,5\$ (75,000 / 1,000 x 0.35), soit environ 26€, un mois après la période de 90 jours initiale.

Des services web spécialisés de l'API Google Maps ont des limites additionnelles :

- « Directions » - fournit les directions au format texte - limitées à 2 500 par jour.
- « Distance » - fournit la distance et le temps de trajet - limitée à 100 éléments par requête et 2 500 par jour.
- « Relief » – altitude à certains endroits - limité à 2 500 requêtes par jour et chaque requête ne peut excéder la fourniture de plus de 512 altitudes.
- « Geocoding » - géocodage, convertit une adresse de rue en latitude et longitude - limité à 2 500 par jour.
- « Places » - retourne les établissements et différents points d'intérêt dans une zone - requiert une clef API et est limité à 1 000 requêtes par jour.

Une clef API Google Maps est un code numérique qui enregistre votre site auprès de Google. Si cela n'est pas nécessaire pour des applications normales, il faudrait s'y intéresser si les limites d'utilisation sont dépassées ou si le service « Places » est utilisé.

L'exemple dans la figure 14.9 montre le code Javascript et les appels à l'API pour afficher une carte simple qui est centrée à un emplacement spécifique. Le niveau de zoom, qui peut aller de 0 à 21, est paramétré à 15 sous *myOptions*. Le centre est défini avec des valeurs spécifiques de latitude et de longitude, et l'option ROADMAP est sélectionnée pour définir le style de carte. Tous les appels de l'API sont générés dans la fonction initialize. Cette fonction est appelée onload au sein du <body> du fichier HTML.

```
<html>
<head>
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false">
</script>
<script type="text/javascript">
function initialize() {
var latlng = new google.maps.LatLng(46.810928,
90.817981);
var myOptions = {
zoom: 15,
center: latlng,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map=new
google.maps.Map(document.getElementById("map_canvas"),
myOptions);
</script>
</head>
<body onload="initialize">
<div id="map_canvas" style="width:600px; height:300px">
</div>
</body>
</html>
```

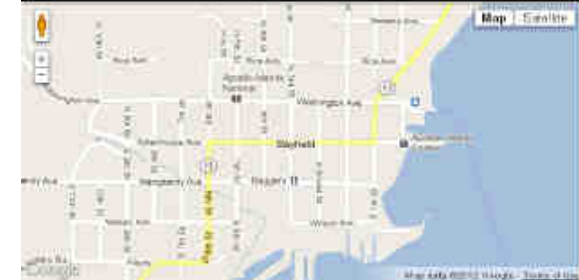


Figure 14.9. Requête de carte Google basique. Le centre de la carte peut être modifié ainsi que le niveau de zoom et le type de carte (© 2014 Google)

Un simple changement peut être apporté à ce code en donnant de nouvelles valeurs de latitude et de longitude. Pour déterminer la latitude/longitude d'un point spécifique, il est possible d'appliquer plusieurs méthodes :

- Dans Google Maps avec un clic droit (control + clic sur Mac), sélectionner "Plus d'infos sur cet endroit" et les données apparaîtront en haut à gauche de la fenêtre Google Maps.

- Un clic droit avec MapQuest affiche la valeur dans une fenêtre pop-up.
- Dans Bing Maps, la latitude et la longitude sont affichées avec un clic droit.
- Pour afficher les coordonnées au format de degrés décimaux avec Google Earth, il faut sélectionner Outils/Options et cliquer sur l'option degrés décimaux.
- Pour finir, il y a également un certain nombre d'outils en ligne ; rechercher "Trouver latitude et longitude" vous amènera à un site. La plupart de ces sites utilisent Google Maps, comme l'exemple qui suit :

<http://findlatitudeandlongitude.com>.

Un autre changement basique d'une carte Google est le type ou le style de carte qui est affiché. Google offre quatre vues :

- **MapTypeId.ROADMAP** displays the affiche la carte routière par défaut.
- **MapTypeId.SATELLITE** affiche les images satellites Google Earth.
- **MapTypeId.HYBRID** affiche un mélange entre carte normale et images satellites.
- **MapTypeId.TERRAIN** affiche une carte physique basée sur les informations du terrain.

Le zoom initial peut également être modifié. Une valeur de "0" afficherait une carte dézoomée, de petite échelle. Lorsque le niveau de zoom augmente, l'échelle de carte fait de même. Les valeurs maximales varient d'un endroit à un autre du monde. Généralement, vingt niveaux de détail sont toujours disponibles. Certaines parties du monde ont plus de vingt niveaux de zoom.

14.4 Point, ligne, zone et mélanges de calques

Toutes les cartes sont composées de points, lignes et surfaces. De plus, de nombreuses cartes peuvent être combinées en calques individuels, une fonction qui est la

base des systèmes d'information géographique. Les exemples dans cette section présentent comment ces éléments peuvent être implémentés dans une carte Google.

14.4.1 Points

Le marqueur par défaut de Google est un symbole de goutte d'eau à l'envers, mais un grand nombre d'autres icônes sont disponibles. Il est également possible de concevoir des symboles car ce sont de simples images PNG de 32x32 pixels. Les marqueurs peuvent être statiques ou interactifs. Le principal type de marqueur interactif est le marqueur cliquable.



Figure 14.10. Exemple de marqueur cliquable. La variable *contentString* est définie dans le HTML (*©2014 Google*)

Dans l'exemple de la figure 14.10, la variable *contentString* est définie avec du texte formaté en HTML. Elle est associée à une variable *infoWindow* qui

est par la suite associée avec **google.maps.event.addListener**. Quand un utilisateur clique sur le marqueur, le texte est affiché dans une bulle pop-up. Le HTML pour cette bulle pourrait comporter une image ou même une vidéo en utilisant les balises **img** ou **embed**.

Lorsque de nombreux points doivent être cartographiés et fréquemment mis à jour, un format web appelé Really Simple Syndication (RSS) est souvent utilisé. Le RSS a de nombreux avantages. Les éditeurs de flux RSS ont l'avantage de diffuser le contenu automatiquement tandis que les consommateurs peuvent profiter d'informations cartographiques mises à jour continuellement. Un format de fichier standardisé permet à l'information d'être publiée une seule fois et vue à l'aide de nombreux programmes.

KML, Keyhole Markup Language, est un format utilisé pour décrire des espaces à deux - et trois - dimensions ; il était à l'origine développé pour Google Earth. Il s'agit à présent d'un standard open-source officiellement appelé OpenGIS KML Encoding Standard (OGC KML) et il est entretenu par l'Open Geospatial Consortium (OGC). Ce format spécifie des fonctionnalités telles que les repères, les images, les polygones et les modèles 3D. Les emplacements sont toujours spécifiés avec la latitude et la longitude. De nombreux fichiers KML sont disponibles sur Internet.

La fonction **google.maps.KmlLayer** fonction lit un flux RSS formaté KML et est spécifiée avec une adresse HTTP. Les cartes réalisées de cette façon sont généralement affichées très rapidement. L'inconvénient réside dans le manque de contrôle de l'aspect de la carte puisque les symboles sont définis dans le fichier KML.

L'exemple de la figure 14.11 présente une application d'un flux RSS pour la représentation de tremblements de terre. Ce flux KML particulier est mis à jour quotidiennement et montre les tremblements de terre des sept derniers jours. Chaque marqueur est cliquable et fournit de plus amples informations sur le tremblement de terre. Le code montre

comment faire une carte pour une certaine partie du monde.

Earthquakes in the past week



```
var georssLayer = new
google.maps.KmlLayer('http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M2.5.xml');
Earthquakes in the past week
```

```
var ctaLayer = new var ctaLayer = new
google.maps.KmlLayer('http://earthquake.usgs.gov/earthquakes/catalogs/eqs7day-M2.5.xml', {preserveViewport:true});
ctaLayer.setMap(map);
```

Figure 14.11. Flux RSS défini au format KML, par la United States Geological Survey. Chaque icône localise le tremblement de terre et, après avoir cliqué, décrit l'événement. La carte du bas est affichée grâce à un appel unique vers un calque KML qui ignore le centre défini et le niveau de zoom, et duplique ainsi la plupart du monde. La carte du haut est affichée avec l'option {preserveViewport: true} et ctaLayer qui applique le centre et le niveau de zoom définis par l'utilisateur (© 2014 Google)

14.4.2 Lignes

La fonction Polyligne de Google Maps est utilisée pour dessiner des lignes avec l'API Google Maps. Dans la figure 14.12, la fonction Polyligne connecte les points qui ont été définis au préalable. Les options pour contrôler l'apparence de la ligne incluent strokeColor, strokeOpacity, et strokeWeight. Comme toujours, un centre et un niveau de zoom appropriés doivent aussi être définis. Le centre pourrait être le point central de la ligne elle-même.



Figure 14.12. Ligne créée à partir de trois éléments composés de quatre points. (© 2014 Google)

La distance la plus courte entre deux points apparaît courbe sur la carte puisqu'elle a été calculée avec une terre sphérique. Sur la plupart des projections, incluant celle de Mercator qui est utilisée par tous les services majeurs de cartographie en ligne, la distance la plus courte sur le sol est représentée par une courbe - donnant l'impression qu'il s'agit d'une distance bien plus longue entre les deux endroits. Le grand cercle est défini comme la plus courte distance entre deux points qui divise la Terre en deux hémisphères égaux. C'est une fonctionnalité supportée dans l'API Google Maps à l'aide de l'option polyligne geodesic: true (voir fig. 14.13).

```
var flightPlanCoordinates = [
new google.maps.LatLng(37.772323, -
122.214897),
new google.maps.LatLng(21.291983, -
157.821656),
new google.maps.LatLng(-18.142659, 178.431),
new google.maps.LatLng(-17.46758, 159.02782)
];
var flightPath = new google.maps.Polyline({
path: flightPlanCoordinates,
strokeColor: "#FF0000",
strokeOpacity: 1.0,
strokeWeight: 3
```



Figure 14.13. Option polyligne geodesic: true connectant deux points au sein du grand cercle, la distance la plus courte entre deux endroits sur la sphère. Cela donne l'impression d'une ligne plus longue sur la carte à cause de la projection - (© 2014 Google)

14.4.3 Surfaces

Un polygone peut être vu comme une ligne qui se ferme sur elle-même. Il consiste en une série de points, le dernier point étant toujours égal au premier. Les deux attributs additionnels qui ont besoin d'être définis pour google.maps.Polygon sont l'ombrage et l'opacité de l'intérieur de la surface.

```

<script type="text/javascript">
function initialize() {
var myLatLng=new
google.maps.LatLng(24.886436490787712,-
70.2685546875);
var myOptions = {
zoom: 5,
center: myLatLng,
mapTypeId: google.maps.MapTypeId.TERRAIN
};
var map = new
google.maps.Map(document.getElementById("map_c
anvas"),
myOptions);
var triangleCoords = [
new google.maps.LatLng(25.774252, -80.190262),
new google.maps.LatLng(18.466466, -66.118292),
new google.maps.LatLng(32.321384, -64.75797),
new google.maps.LatLng(25.774252, -80.190262)
];
var bermudaTriangle = new
google.maps.Polygon({
paths: triangleCoords,
strokeColor: "#FF0000",
strokeOpacity: 0.8,
strokeWeight: 2,
fillColor: "#FF0000",
fillOpacity: 0.35
});
bermudaTriangle.setMap(map);
}
</script>

```



Figure 14.14. La fonction Polygone de Google trace une forme fermée. Les options incluent strokeColor, strokeOpacity, strokeWeight, fillColor et fillOpacity (© 2014 Google)

La figure 14.14 représente le Triangle des Bermudes dans l'océan Atlantique. Quatre points sont définis pour représenter le triangle. Ces points sont chargés dans un

tableau nommé triangleCoords. Ce tableau passe ensuite vers google.maps.Polygon. Les paramètres incluent strokeColor, strokeOpacity, stroke-Weight, fillColor and fillOpacity.

14.4.4 Calques

Jusqu'ici, nous avons superposé les points, lignes, et zones qui étaient définis comme vecteurs de latitude et de longitude. Maintenant, nous superposons une image ou une photo qui peut être une photo aérienne, une image satellite ou une image scannée. L'intérêt de superposer une image est que cette superposition peut être effectuée rapidement. Aucune conversion ni aucun traçage majeur n'est nécessaire pour placer l'information puisque la carte sous-jacente est dans le même format. Les fichiers matriciels peuvent être superposés comme entités simples ou divisés en dalles pour correspondre précisément aux dalles de la carte sous-jacente.

L'exemple de la figure 14.15 présente une carte qui a été scannée et enregistrée au format JPEG. La latitude et la longitude des coins sud-est et nord-est ont été estimées et ensuite définies en utilisant imageBounds. Ces coordonnées sont combinées avec l'adresse de l'image dans l'objet Oldmap.

```

function initialize() {
var newark = new google.maps.LatLng (40.740,-
74.18);
var imageBounds = new
google.maps.LatLngBounds(
new google.maps.LatLng (40.712216,-74.22655),
new google.maps.LatLng (40.773941,-74.12544));
var myOptions = {
zoom: 12,
center: newark,
mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map=new
google.maps.Map(document.getElementById("map_c
anvas"),myOptions);
var oldmap = new google.maps.GroundOverlay(
"http://www.lib.utexas.edu/maps/historical/newark_nj_192
2.jpg",
imageBounds);
oldmap.setMap(map);
}

```

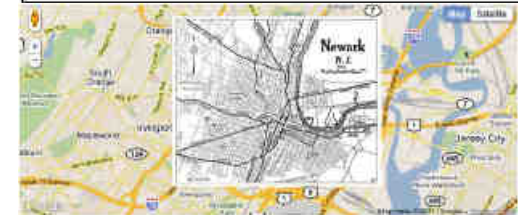


Figure 14.15. Superposition d'une carte scannée au format JPEG. Carte de Newark, dans le New Jersey, avec l'autorisation des University of Texas Libraries, et de l'University of Texas à Austin. (© 2014 Google)

14.5 Cartographie mobile

Les appareils de géolocalisation se démocratisent de plus en plus. Aujourd'hui, les téléphones portables peuvent quasiment tous être localisés à quelques mètres près. Les smartphones ont également la possibilité d'afficher leur emplacement courant sur une carte. Les tablettes basées sur iOS de Apple et Android de Google peuvent généralement faire de même, avec l'avantage d'afficher une image beaucoup plus grande.

Il y a de nombreux types différents d'appareils mobiles et de nombreuses façons de déterminer une position. Afin de proposer une approche standardisée, le World Wide Web Consortium (W3C) a créé une API de géolocalisation disponible librement. Supportée par pratiquement tous les navigateurs, l'API utilise plusieurs méthodes pour trouver l'emplacement d'un ordinateur ou d'un appareil mobile (Svennerberg 2010, p. 235).

Le Global Positioning System (GPS) est une méthode pour déterminer une position, mais il fonctionne uniquement si la vue du ciel n'est pas obstruée. Dans les zones urbaines, la méthode la plus répandue pour déterminer une position est la triangulation basée sur le Wi-Fi et les antennes de réseau téléphonique. Un logiciel de localisation développé par une entreprise basée à Boston, Skyhook, utilise un réseau massif de références comprenant les emplacements de plus de 250 millions de points d'accès Wi-Fi et d'antennes téléphoniques. Afin de développer cette base de données, Skyhook a déployé des enquêteurs pour surveiller chaque route, autoroute et chemin dans des dizaines de milliers de villes dans le monde entier, scannant les points d'accès Wi-Fi et les antennes téléphoniques et enregistrant précisément leur emplacement.

La figure 14.16 montre comment est présentée la localisation d'un appareil sur une carte via un navigateur utilisant l'API W3C. La déclaration

`“navigator.geolocation.getCurrentPosition (function(position)”`

donne la position courante de l'appareil. Si la position ne peut pas être déterminée par GPS, l'API utilise la méthode de triangulation basée sur un réseau Wi-Fi. Cet exemple présente une fenêtre d'informations sur la position courante.

L'exemple de la figure 14.16 présente un marqueur cliquable.

Le contentString de la bulle de texte affiche la latitude et la longitude.



```

if(navigator.geolocation) {
  browserSupportFlag = true;
  navigator.geolocation.getCurrentPosition(function(position) {
    initialLocation = new google.maps.LatLng(
      position.coords.latitude, position.coords.longitude);
    contentString = "Pt:"
      + "position.coords.latitude+"
      + "position.coords.longitude ";
    map.setCenter(initialLocation);
    infoWindow.setContent(contentString);
    infoWindow.setPosition(initialLocation);
    var marker = new google.maps.Marker({
      position: initialLocation,
      map: map,
      title: "Hello World!"
    });
    google.maps.event.addListener(marker, 'click',
      function() {
        infoWindow.open(map, marker);
      });
  }, function() {
    handleNoGeolocation(browserSupportFlag);
  });
}
}

```

Figure 14.16. Texte affiché. Dans la bulle cliquable, on retrouve la latitude et la longitude du point actuel de l'appareil mobile. Le "+" dans la déclaration contentString est utilisée pour concaténer les nombres dans une chaîne de texte. (© 2014 Google)

14.6 Conclusions

Nous vivons dans une époque fabuleuse pour la cartographie. En vingt ans, entre les années 1970 et 1990, les cartes sont passées d'objets statiques sur papier à des présentations interactives fournies par un réseau électronique. Depuis ces années, les cartes sont devenues encore plus interactives puisqu'elles sont capables de fournir des informations - des informations thématiques sous forme de superpositions de calques, mais aussi la modification de la carte de base sous-jacente. Les exercices de ce chapitre ont fait office d'introduction au nouveau monde de la cartographie via Internet. Les outils qui ont été présentés peuvent être utilisés pour créer des types de cartes très sophistiqués.

Bibliographie

Google Maps JavaScript API V3 Basics (2011). (Rechercher : Google Maps JavaScript API V3 Basics).

Neumann, A., Winter A. M. (2003), "Web-mapping with Scalable Vector Graphics (SVG) : Delivering the promise of high quality and interactive web maps." In: Peterson, M. P. (ed.) *Maps and the Internet*. Elsevier, Amsterdam, pp.197–220.

Peterson MP (2008), *International Perspectives on Maps and the Internet*. Springer, Berlin.

Svennerberg, Gabriel (2010), *Beginning Google Maps API 3*. New York, NY: Apress.

W3Schools.com (2011). JavaScript Tutorial. <http://www.w3schools.com/js/default.asp>. (search: Learning JavaScript).

Willard, Wendy (2009), *HTML: A Beginner's Guide*. Berkeley, CA: Osborne/McGraw-Hill.

Note : Le contenu de ce chapitre est basé sur le livre de l'auteur intitulé *Mapping in the Cloud* et publié par Guilford Press.