

# **Cartographie Geolocalisation et capteurs**

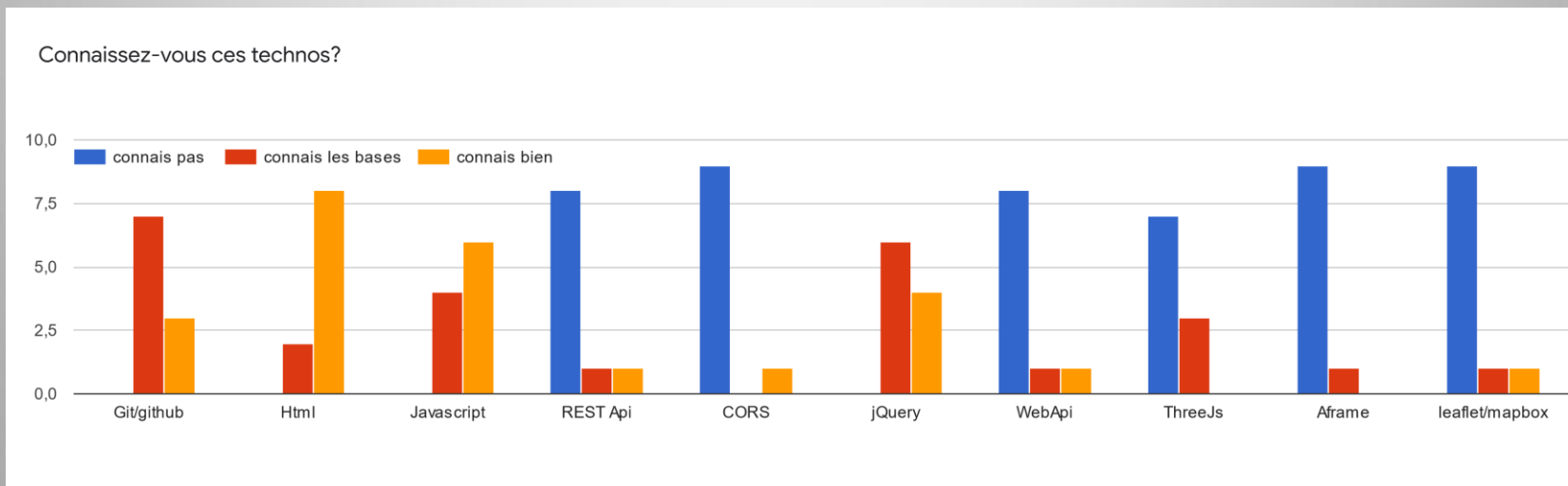
<https://github.com/vestri/CoursGeo>

**Christophe Vestri**

Le mercredi 3 janvier 2023

# Plan du cours

- 3 janvier : Intro, github, Capteur/Geoloc en HTML5
- 10 janvier: carto/geo, leaflet/mapBox, rest Api
- 24 janvier: 2D/3D: Canvas, WebGL et Three.js
- 31 janvier: Aframe/AR.js, exercice + projet
- 7 février : Projets



# Objectifs du cours

- Bases de geolocalisation et de la cartographie
- Expérimenter quelques méthodes et outils web geo/3D
- Réaliser un projet

<https://github.com/vestri/CoursGeo>

- Evaluation:
  - Exos des cours (50%)
  - Projet (50%)

# Projet final

- **Projet final**
  - Capteurs mouvement/orientation
  - GéoLocalisation et/ou objets geolocalisés
  - UI et scene 3D, interaction
  - Exemples:
    - Compas 2D/3D: carte 2D + geoloc et directions 3D
    - Objets 3D animés avec interaction smartphone
    - Réalité augmentée (HtIm5/JS)

# Plan Cours 1

- Géolocalisation
- Référentiels et modèles de projection
- Capteurs smartphones
- Git/Github/Github.io
- Exercices d'accès capteurs smartphone
  - Geolocalisation
  - Device Events
  - UI et touch events

# Mon parcours

**Christophe Vestri**

[vestri@3DVTech.com](mailto:vestri@3DVTech.com)

DUT-Ingénieur-DEA-Thèse



AIRBUS



IMRA

**3DVTech**

- Développement traitement image
- Bureau d'étude et conseil



3DVTech

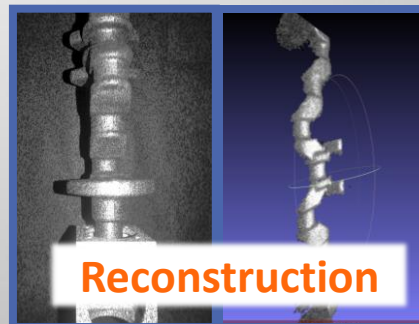
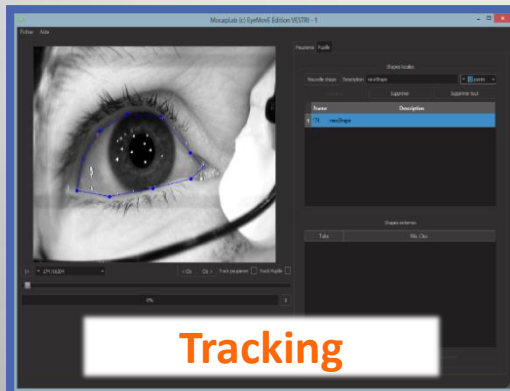
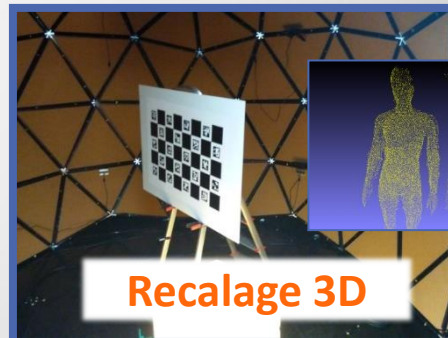
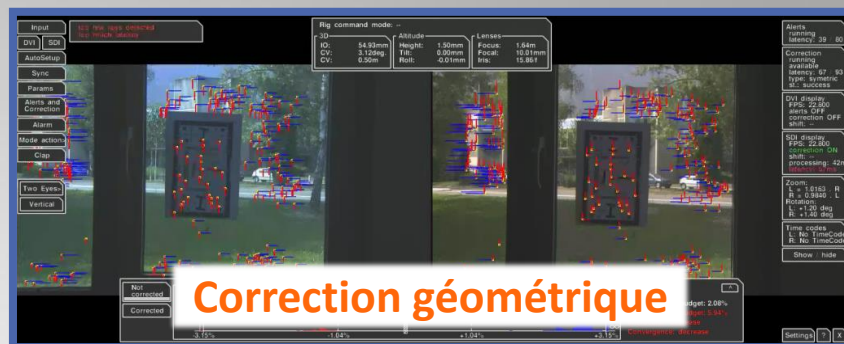
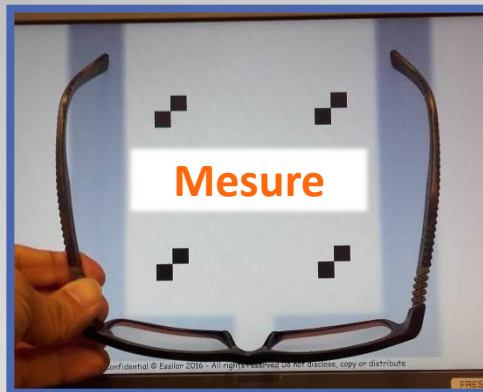
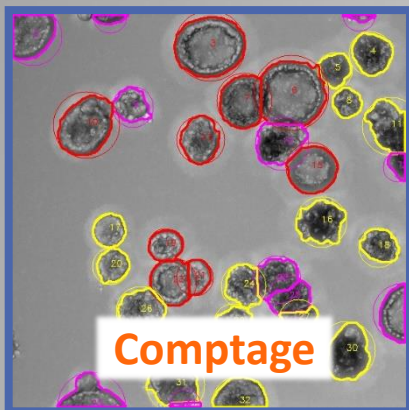
[www.3DVTech.com](http://www.3DVTech.com)

**R&D Vision**

Expert traitement d'images





R&D  
VISION  
[www.rd-vision.com](http://www.rd-vision.com)





# Vous





Choisir la forme :

Sphere

Choisir la couleur de la forme :

orange

Entrer les paramètres de la forme :


Valeur 1 :	<input type="text" value="0.5"/>
Valeur 2 :	<input type="text" value="15"/>
Valeur 3 :	<input type="text" value="15"/>
Valeur 4 :	<input type="text" value="8"/> <input type="button" value="Valider"/>

Les informations de votre figure

Sylvestre

orange

0.5 15 15 8





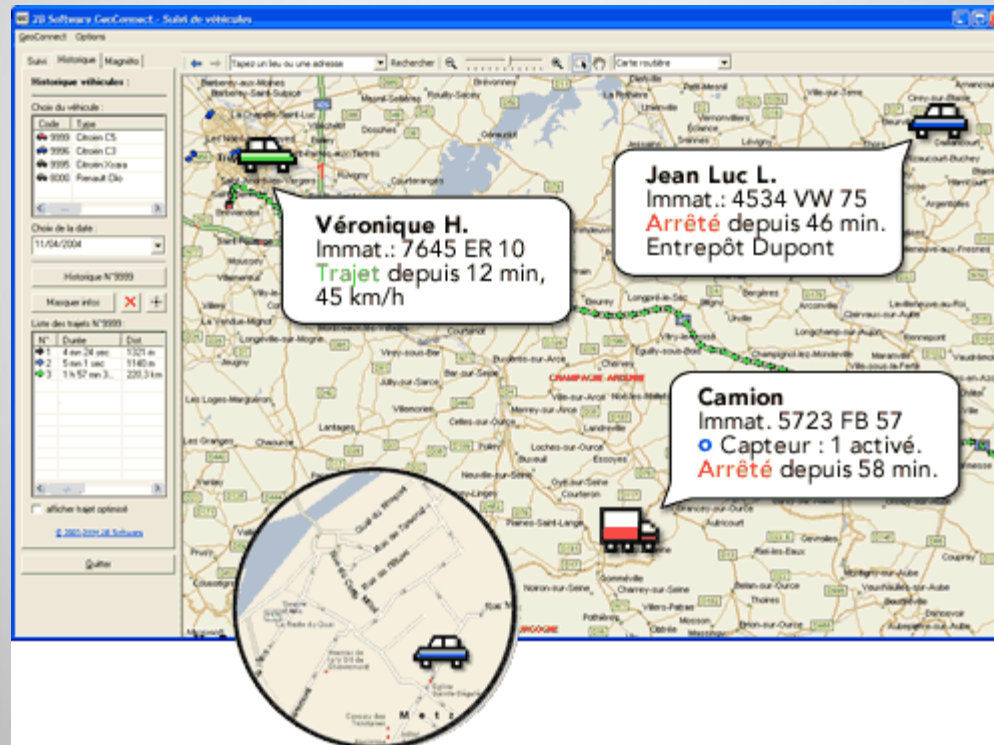
# Plan Cours 1

- **Géolocalisation**
- Référentiels
- Git/Github/Github.io
- Capteurs smartphones
- Exercices d'accès capteurs smartphone
  - Geolocalisation
  - Device Events
  - UI et touch events

# Définitions

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

La géo-localisation est une technologie qui permet de collecter des informations permettant de localiser un objet ou une personne sur une carte, à l'aide de coordonnées géographiques.



# Géolocalisation

- **Géolocalisation**
- Référentiels
- Git
- Capteurs
- Exos

## Techniques de [géolocalisation](#)

- Localisation par satellite,
- Réseaux mobiles GSM/GPRS/UMTS (+/-100m à qqs km)
- Bornes wifi,
- Puces RFID
- Vidéo-surveillance / vidéoprotection,
- Cartes de paiement et de transport.

# Géolocalisation par Satellite

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

## Systèmes de positionnement par satellite

### Systèmes de navigation satellitaires existants ou en développement [\[ modifier \]](#) [modifier le code](#) ]

Les systèmes de positionnement satellitaires avec une couverture globale sont :

- GPS pour les États-Unis (pleinement opérationnel depuis 1995) ;
- GLONASS pour la Russie (opérationnel entre 1996 et 1999, puis de nouveau opérationnel depuis 2010) ;
- Galileo pour l'Europe (opérationnel depuis 2016<sup>1</sup>) ;
- Compass ou Beidou-2 et 3 (évolution à dimension mondiale de Beidou-1, régional) pour la Chine.

Les systèmes de positionnement avec une couverture régionale :

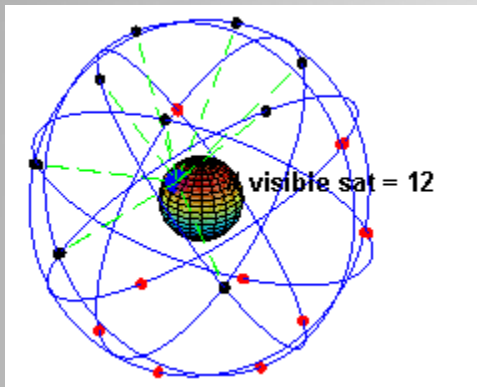
- Beidou-1 pour la Chine ;
- IRNSS pour l'Inde (en cours de déploiement en 2015) ;
- QZSS pour le Japon (en cours de déploiement en 2015).

Comparaison des caractéristiques du segment spatial (2017)

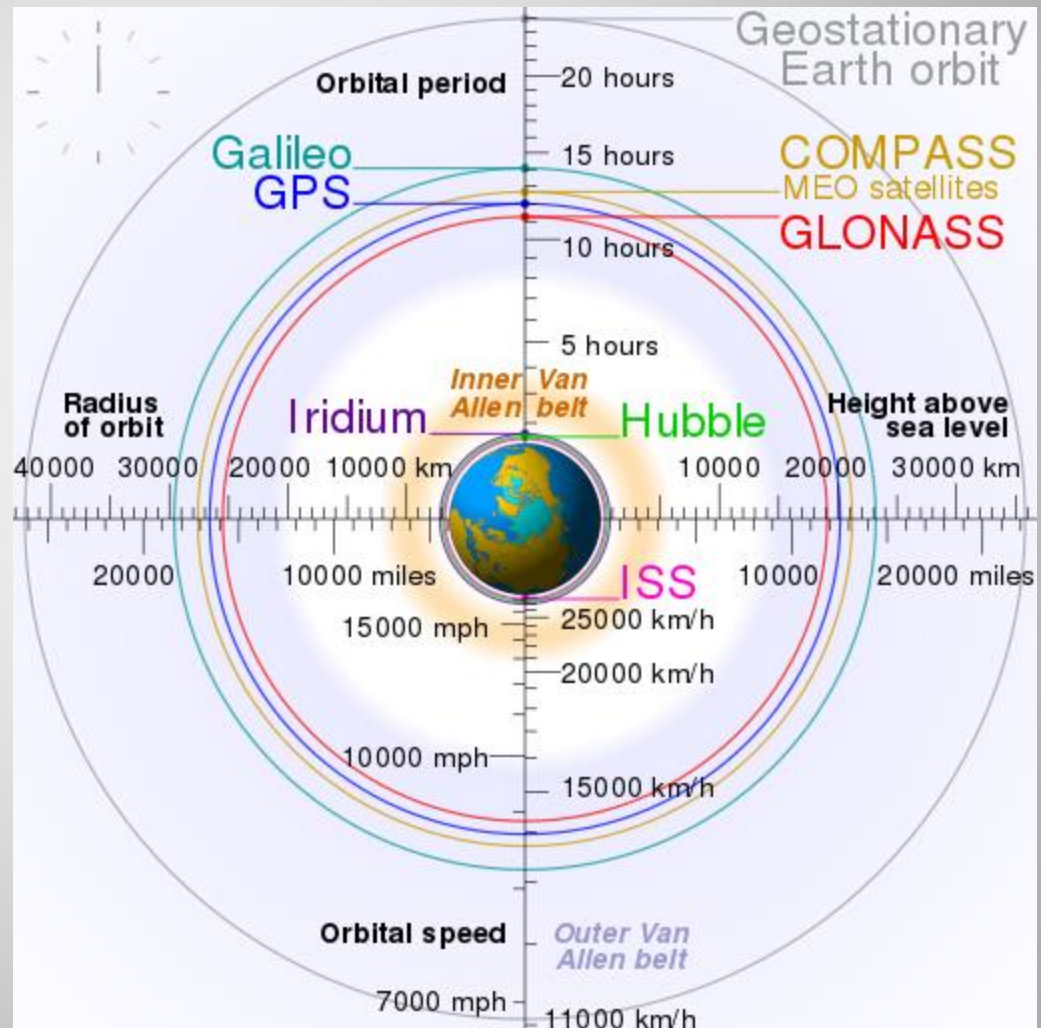
Caractéristique	GPS	GLONASS	GALILEO	Beidou/Compass
Segment spatial				
Altitude	20 200 km	19 100 km	23 222 km	21 528 km
Inclinaison	55°	64,8°	56°	55°
Période orbitale	11 h 58	11 h 15	14 h 07	12 h 53
Nombre de plans orbitaux	6	3	3	3
Nombre de satellites opérationnels (en cible)	31 (31)	24 (24)	15 (27)	20 <sup>2</sup> (27 + 5)

# Géolocalisation par Satellite

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos



Constellation de satellites constituant un système de positionnement par satellites ; ici celle du [système GPS](#).



# Les domaines d'application

- **Géolocalisation**
- Référentiels
- Git
- Capteurs
- Exos

## L'information, le tourisme

La géo-localisation permet de faciliter l'accès à l'information qui vous entoure au travers d'applications de réalité augmentée et autres.



## Les transports

Aujourd'hui, de plus en plus de titres de transport contiennent une puce électronique avec nos informations. Cela permet de gagner du temps. Par traitement d'information, on connaît mieux vos déplacements habituels et leurs fréquences.

# Les domaines d'application

- **Géolocalisation**
- Référentiels
- Git
- Capteurs
- Exos

## **La recherche**

Des chercheurs américains ont mis au point un système qui utilise la géo-localisation et qui est capable de déterminer votre âge, votre sexe et vos goûts.

## **Le marketing (geomarketing)**

La géo-localisation peut permettre aux marques d'augmenter leurs ventes. En effet, elles peuvent nous repérer et, en fonction de l'endroit où l'on se trouve, nous proposer des bons plans à ne pas rater, des services ou des produits adaptés.

Gestion des campagnes de couponing par geo-localisation

## **Le traçage (geotracking)**

Traçage d'une flotte de véhicule, jusqu'au traçage des personnes (geofamily)

## **Les rencontres géographiques**

Application de rencontres perso, pro, commerciales



# Les limites

- **Géolocalisation**
- Référentiels
- Git
- Capteurs
- Exos

« Notre liberté se bâtit sur ce qu'autrui ignore de nos existences »

Que deviennent les informations collectées sur nos déplacements ?

L'article 9 du code civil : « chacun a droit au respect de sa vie privée »

La géo-localisation n'a pas besoin systématiquement de garder une trace de nos déplacements

Obligation d'une géo-localisation volontaire de la part de l'utilisateur.

# Les limites

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

## Des règles simples pour un meilleur respect des libertés individuelles des utilisateurs

Intérêts mutuels à être transparent vis-à-vis de l'utilisateur.

Volonté des utilisateurs à être maîtres de leurs informations personnelles

De ces constats, deux règles doivent être observées afin d'éviter des problèmes

- Toute application utilisant la géo-localisation, doit adopter le principe dit « check in »
- Déclarer un fichier client à la [CNIL](#)

# Plan Cours 1

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

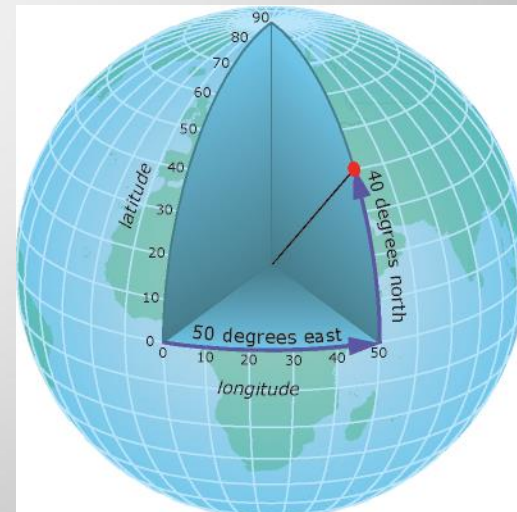
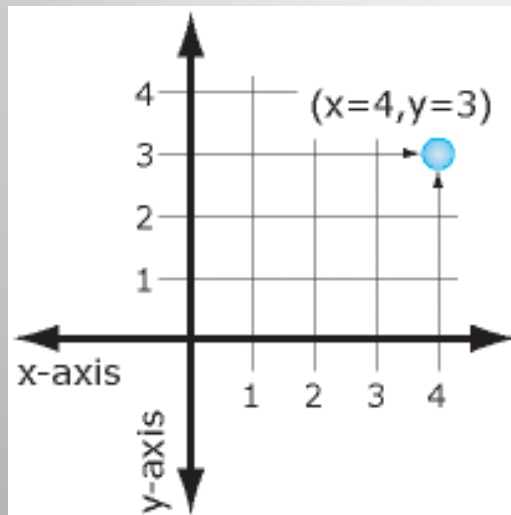
- Géolocalisation
- **Référentiels et modèles de projection**
- Git/Github/Github.io
- Capteurs smartphones
- Exercices d'accès capteurs smartphone
  - Geolocalisation
  - Device Events
  - UI et touch events

# Les Systèmes de Coordonnées de Référence

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

Les SCR (Systèmes de Coordonnées de Référence) sont des modèles mathématiques permettant, grâce aux coordonnées, de faire le lien entre un endroit réel sur terre et sa représentation en plan.

En général, les SCR se divisent en systèmes de coordonnées de référence projetées (aussi appelés systèmes de coordonnées de référence cartésiennes ou rectangulaires) et systèmes de coordonnées de référence géographique.



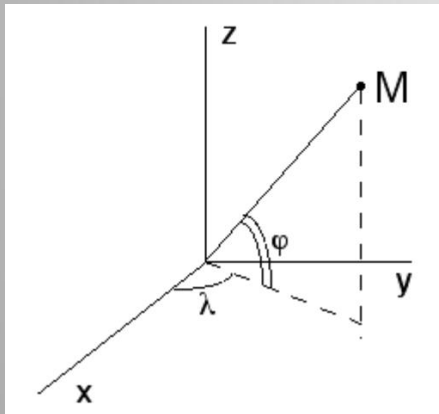
# Systèmes Géographiques et Cartographiques

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

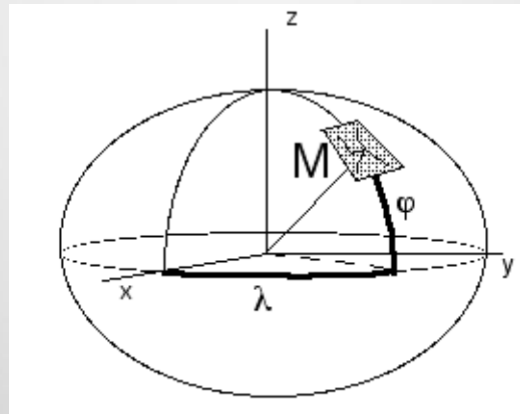
- Construction d'un référentiel géographique

Choix d'un ellipsoïde

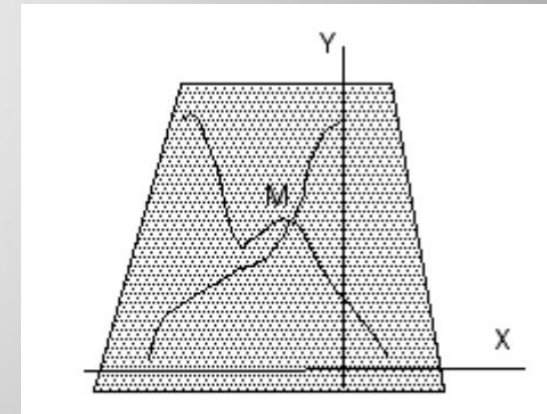
Choix d'une projection



Système cartésien  
 $x, y, z$



Système géographique  
 $\varphi, \lambda, z$

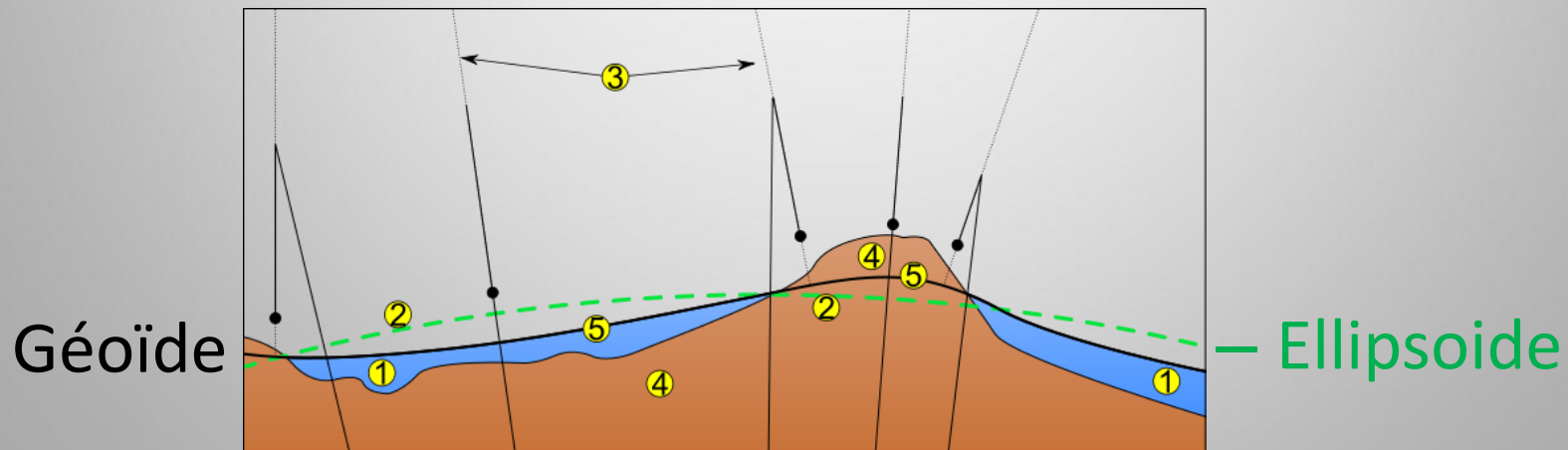
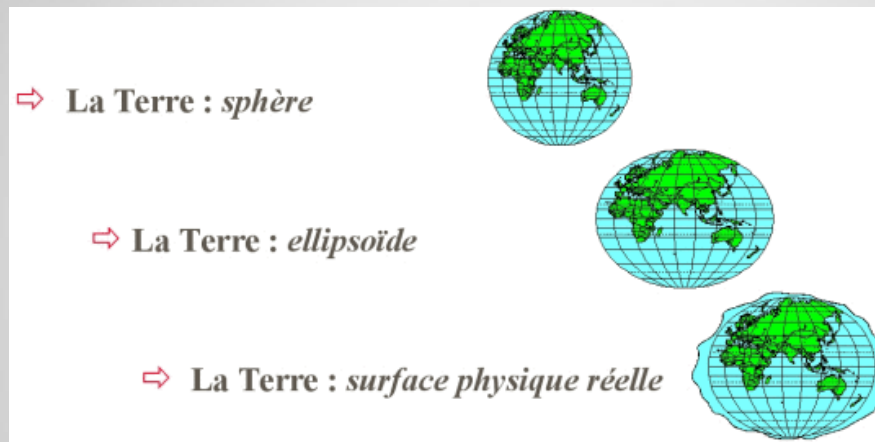


Système cartographique  
 $X, Y, Z$

# Systemes Géographiques et Cartographiques

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

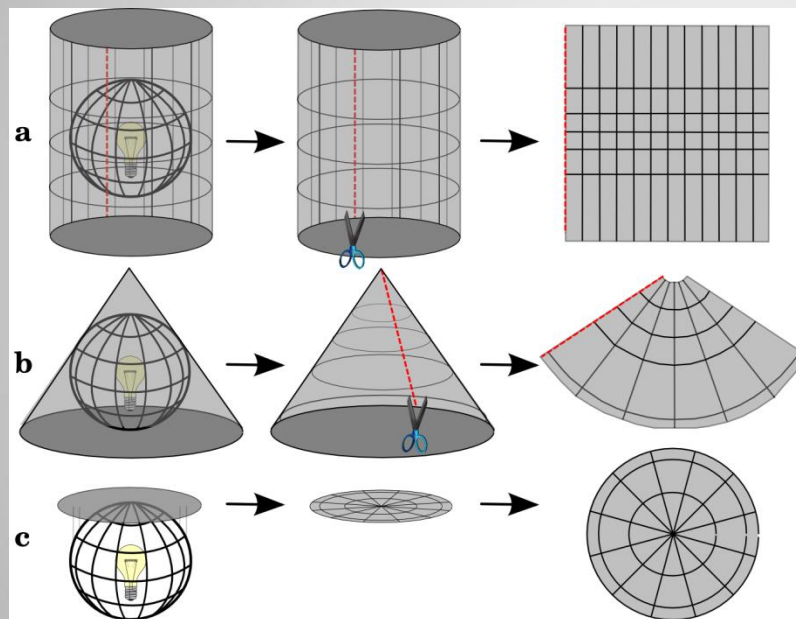
- Représentation de la terre



# Systemes Géographiques et Cartographiques

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

Les 3 familles de projections cartographiques :



- a) Projections cylindriques
- b) Projections coniques
- c) Projections planes ou azimutales

Chaque projection cartographique a des avantages et des désavantages. La meilleure projection d'une carte dépend de l'échelle de la carte, et pour l'objectif pour laquelle elle sera utilisée.

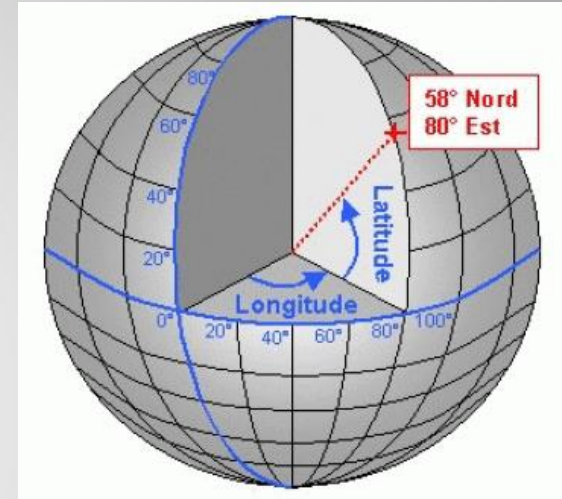


# Systèmes Géographiques et Cartographiques

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

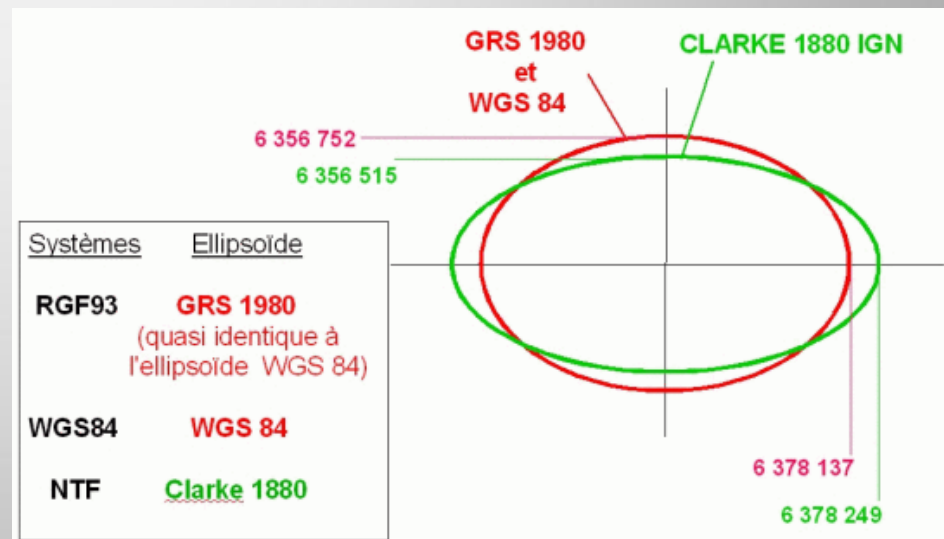
- Un point de la surface terrestre est repéré en fonction d'un ellipsoïde par :

- sa longitude :  $\lambda$  (Lambda)
- sa latitude :  $\phi$  (Phi)



- Différents systèmes:

- GPS ([WGS84](#)),
- Europe ([ETRS89](#))
- France ([NTF](#), [RGF 93](#))



# Systèmes Géographiques et Cartographiques

- Géolocalisation
- **Référentiels**
- Git
- Capteurs
- Exos

- Coordonnées GPS: Lat/Lon

- La salle 202:

- $43.616513, 7.072094 = 43^{\circ}36'59.5''N + 7^{\circ}04'19.5''E$

- Plus d'infos:

- Wikipédia

- IGN: <http://geodesie.ign.fr/index.php> et <http://education.ign.fr/dossiers/mesurer-la-terre>

- <http://seig.ensg.eu/>

- [http://sgcaf.free.fr/pages/techniques/ign\\_coordonnees.htm](http://sgcaf.free.fr/pages/techniques/ign_coordonnees.htm)

# Pour tester sur un Mobile

## Git/Github/Github.io

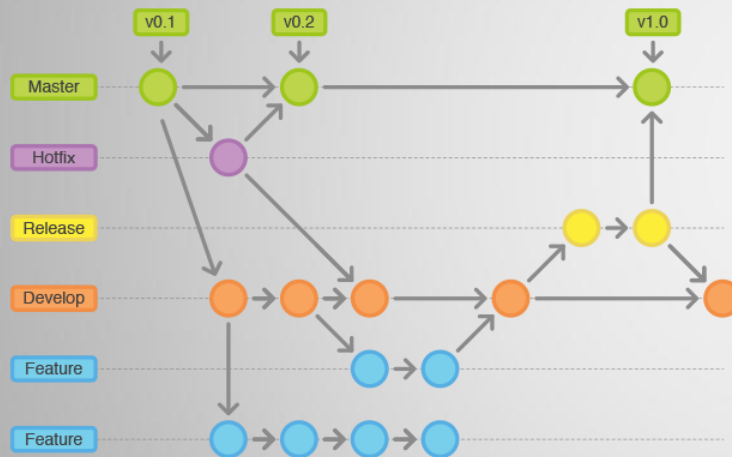
- Géolocalisation
- Référentiels
- **Git**
- Capteurs
- Exos

- Créer un compte sur github
- Poster le code sur github
- Tester avec votre smartphone
- Sinon: <https://codepen.io/> mais il faudra m'envoyer le lien (pas terrible)

# Git

- Géolocalisation
- Référentiels
- **Git**
- Capteurs
- Exos

- Logiciel de versioning créé en 2005 par Linus Torvalds, le créateur de Linux.



## Quelques commandes [\[ modifier \]](#) [\[ modifier le code \]](#)

Git dispose notamment des commandes suivantes :

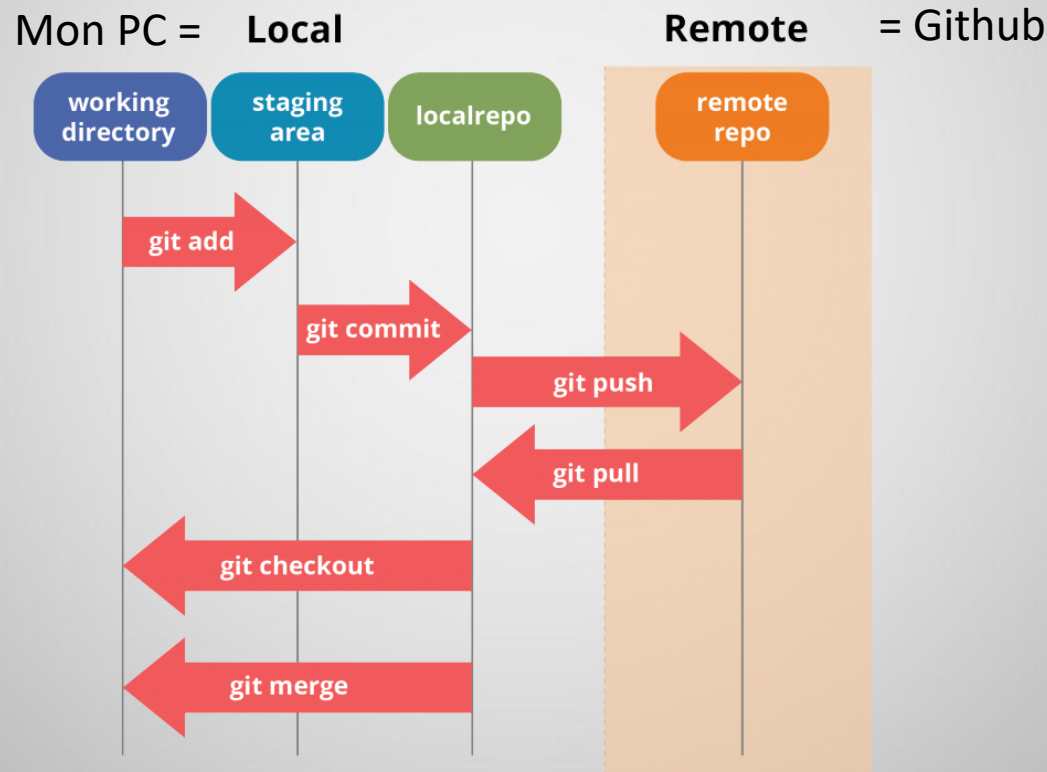
- `git init` crée un nouveau dépôt ;
- `git clone` clone un dépôt distant ;
- `git add` ajoute de nouveaux objets *blobs* dans la base des objets pour chaque fichier modifié depuis le dernier *commit*. Les objets précédents restent inchangés ;
- `git commit` intègre la somme de contrôle *SHA-1* d'un objet *tree* et les sommes de contrôle des objets *commits* parents pour créer un nouvel objet *commit* ;
- `git branch` liste les branches ;
- `git merge` fusionne une branche dans une autre ;
- `git rebase` déplace les commits de la branche courante devant les nouveaux commits d'une autre branche ;
- `git log` affiche la liste des commits effectués sur une branche ;
- `git push` publie les nouvelles révisions sur le *remote*. (La commande prend différents paramètres) ;
- `git pull` récupère les dernières modifications distantes du projet (depuis le *Remote*) et les fusionne dans la branche courante ;
- `git stash` stocke de côté un état non commité afin d'effectuer d'autres tâches.

- Outils: TortoiseGit, SourceTree...

# GitHub

- Géolocalisation
- Référentiels
- **Git**
- Capteurs
- Exos

- Service d'hébergement «gratuit» utilisant git



# Page Github

- Géolocalisation
- Référentiels
- **Git**
- Capteurs
- Exos

- Une page web directement accessible avec smartphone (pour tester)
- Gère les [CORS](#), on mets tous les fichiers nécessaires (images, modèles 3D) en local
- <https://pages.github.com/>

# Premier exercice

<https://github.com/vestri/CoursGeo>

- Géolocalisation
- Référentiels
- **Git**
- Capteurs
- Exos

- Créer votre compte Github
- Installer git ou tortoiseGit (git avec GUI)
- Installer une page pour le projet et les exercices: <https://pages.github.com/>
  - Créer projet, pusher le code, activer la page dans settings en sélectionnant la branche
  - Si projet -> [www.github.com/login/projet](http://www.github.com/login/projet)
  - Si login.github.io -> [login.github.io](http://login.github.io)



# Capteurs smartphones

- Géolocalisation
- Référentiels
- Git
- **Capteurs**
- Exos

- Géolocalisation,
- Accéléromètre,
- Gyromètre
- Magnétomètre,
- Capteurs de pression,
- Capteurs de lumière ambiante,
- Capteur de proximité.

# Applications

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos



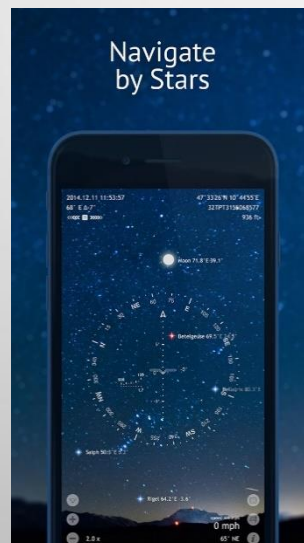
Pokemon Go



Immobilier



GPS

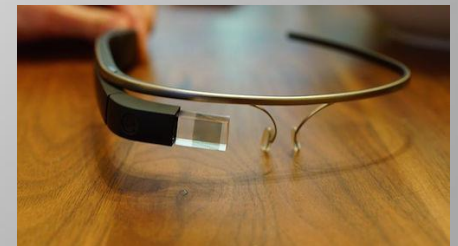


Recherche de points d'intérêts

# RA avec smartphones

- Géolocalisation
- Référentiels
- Git
- **Capteurs**
- Exos

- Smartphones, tout pour la RA
  - Camera + écran – déterminer/montrer ce qui doit être vu
  - Donnée GPS– localisation
  - Compas – quelle direction on regarde
  - Accéléromètre – orientation
  - Connection Internet – fournir des données utiles
- 58% des Français ont un smartphone en 2015
- 90% des 18-24ans
- Lunettes de RA et VR



# RA avec géoloc+sensors

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

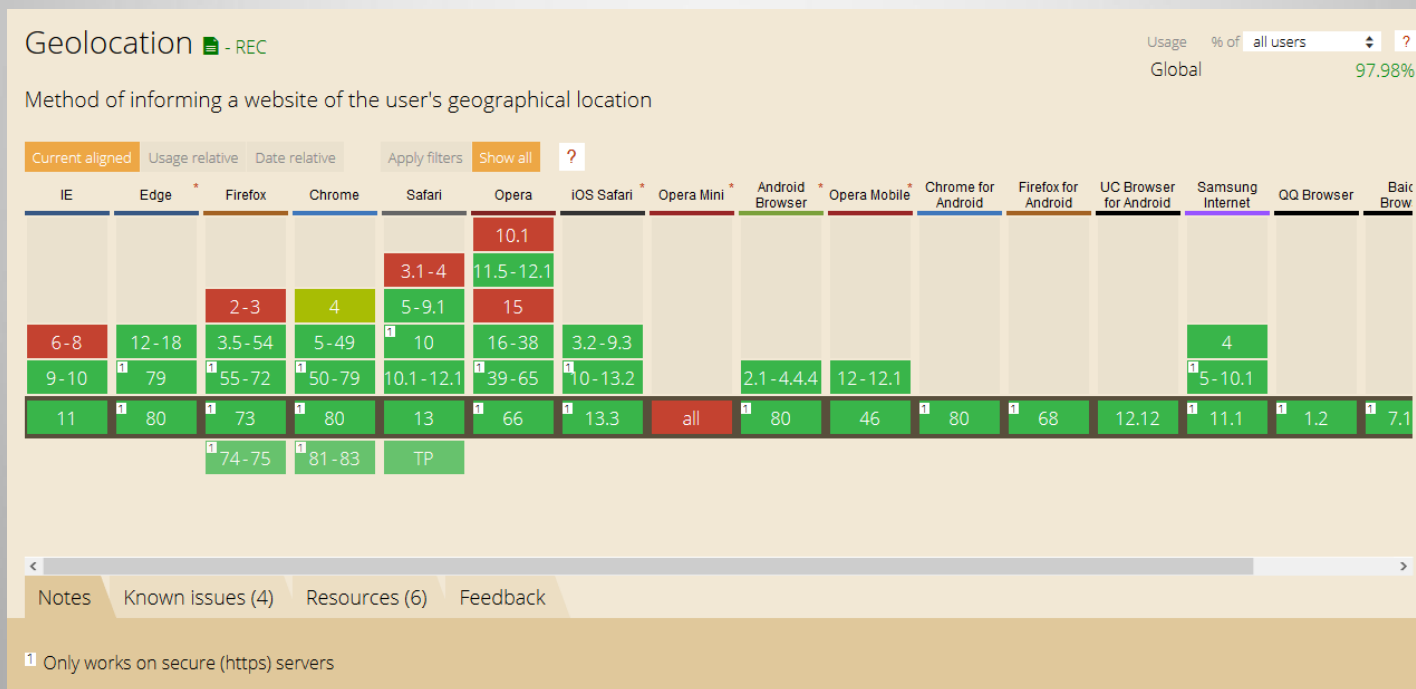
Utilisation des Capteurs du smartphone:

- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité



# Geolocation Specification

- HTML5: Geolocalisation sur mobile
- <https://w3c.github.io/geolocation-api/>
- [Canluse](#): GeoLocation 98%





# Quelques site Html utiles

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

- <https://www.w3schools.com/>
- <https://html-css-js.com/>
- <https://developer.mozilla.org/en-US/>
- <https://caniuse.com/>
- **Arborescence exos**

.git	06/01/2022 21:21	Dossier de fichiers
CartoTD1	06/01/2022 21:20	Dossier de fichiers
Exo1	06/01/2022 21:20	Dossier de fichiers
Exo2	06/01/2022 21:19	Dossier de fichiers
Exo3	06/01/2022 21:19	Firefox HTML Doc...
CartoTD2	06/01/2022 21:21	Dossier de fichiers
Exo1	06/01/2022 21:20	Dossier de fichiers
Exo2	06/01/2022 21:20	Dossier de fichiers
Exo3	06/01/2022 21:19	Dossier de fichiers
CartoTD3	06/01/2022 21:19	Dossier de fichiers
index.html	06/01/2022 21:19	Firefox HTML Doc...

# Outils de debug

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

- En local:
  - `python3 -m http.server`
  - <http://localhost:8000/> firefox ou chrome
- Smartphone android -> Chrome
- <https://developers.google.com/web/tools/chrome-devtools/javascript>
  - Simulation de smartphone (F12)
  - Connecté à un smartphone: <chrome://inspect/>
- Firefox possible ou autres??



# Exercice 2

<https://github.com/vestri/CoursGeo>

- Géolocalisation
- Référentiels
- Git
- Capteurs
- Exos

- **Testez accès Geolocalisation**
- **Afficher**
  - sa position lon, lat
  - la précision de mesure
  - sa vitesse
  - Le time stamp
- **Tester avec/sans gps**
- **utilisez `getCurrentPosition()` et `watchPosition()`**

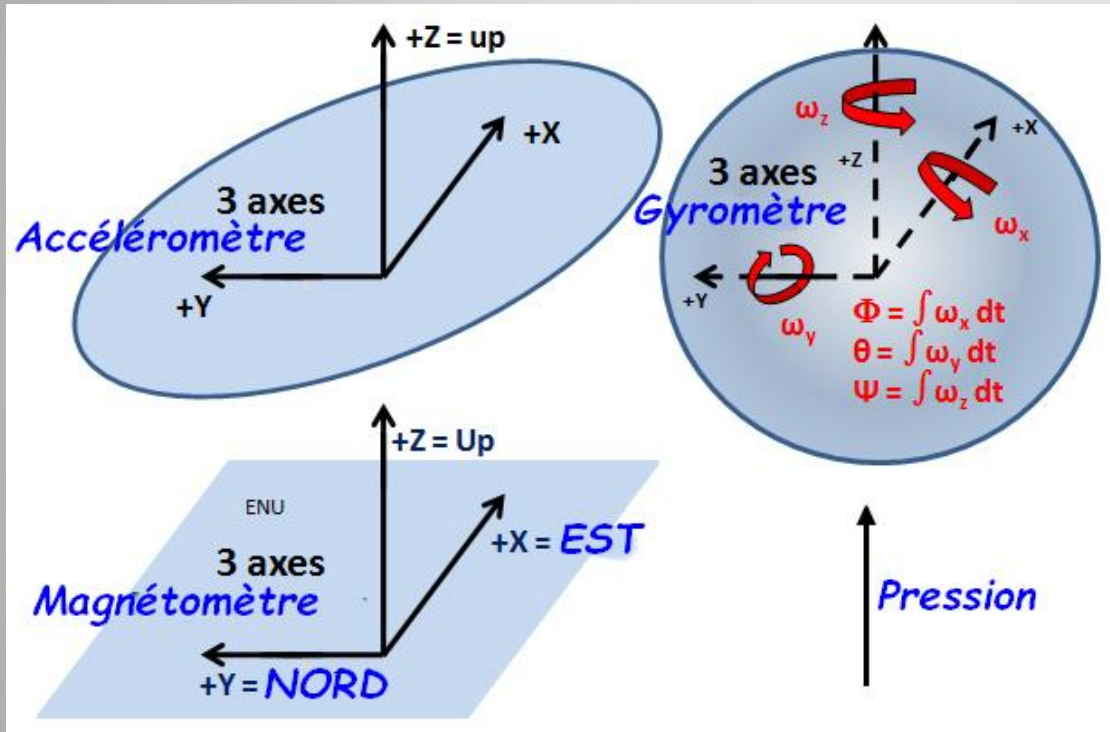
# Capteurs smartphones

- **GPS**
  - Localisation de l'appareil
- **Accéléromètre:**
  - Il ne détecte pas une position, mais une accélération sur chaque X, Y, Z.
  - Permet de savoir dans quelle direction l'appareil se déplace;
- **Gyromètre** (≠ gyroscope qui mesure position angulaire):
  - Le gyromètre ne détecte pas un déplacement linéaire le long d'un axe, mais une accélération de la rotation AUTOUR d'un AXE.
  - Le gyromètre mesure soit des changements dans l'orientation (mouvement angulaires) ou des changements de vitesse de rotation.
- **Magnétomètre**
  - Un magnétomètre mesure les champs magnétiques et parce que la terre possède un champ magnétique significatif,
  - le magnétomètre peut être utilisé comme une boussole.

# Capteurs smartphones

- **Le pedomètre:**
  - Donne le nombre total de pas en 24h, la distance parcourue et l'énergie dépensée.
- **Le capteur d'orientation**
  - Détecte le statut de direction de l'appareil, permet la rotation automatique de l'écran lorsque l'appareil est tourné horizontalement..
- **Le capteur de proximité:**
  - Détecte la présence du corps humain au niveau de l'écouteur de l'appareil.
- **Le Détecteur de luminosité:**
  - Ce capteur permet de savoir quelle est l'intensité lumineuse de l'environnement, ce qui permet de régler automatiquement l'éclairage de l'écran (l'écran consomme beaucoup de courant).

# Capteurs smartphones

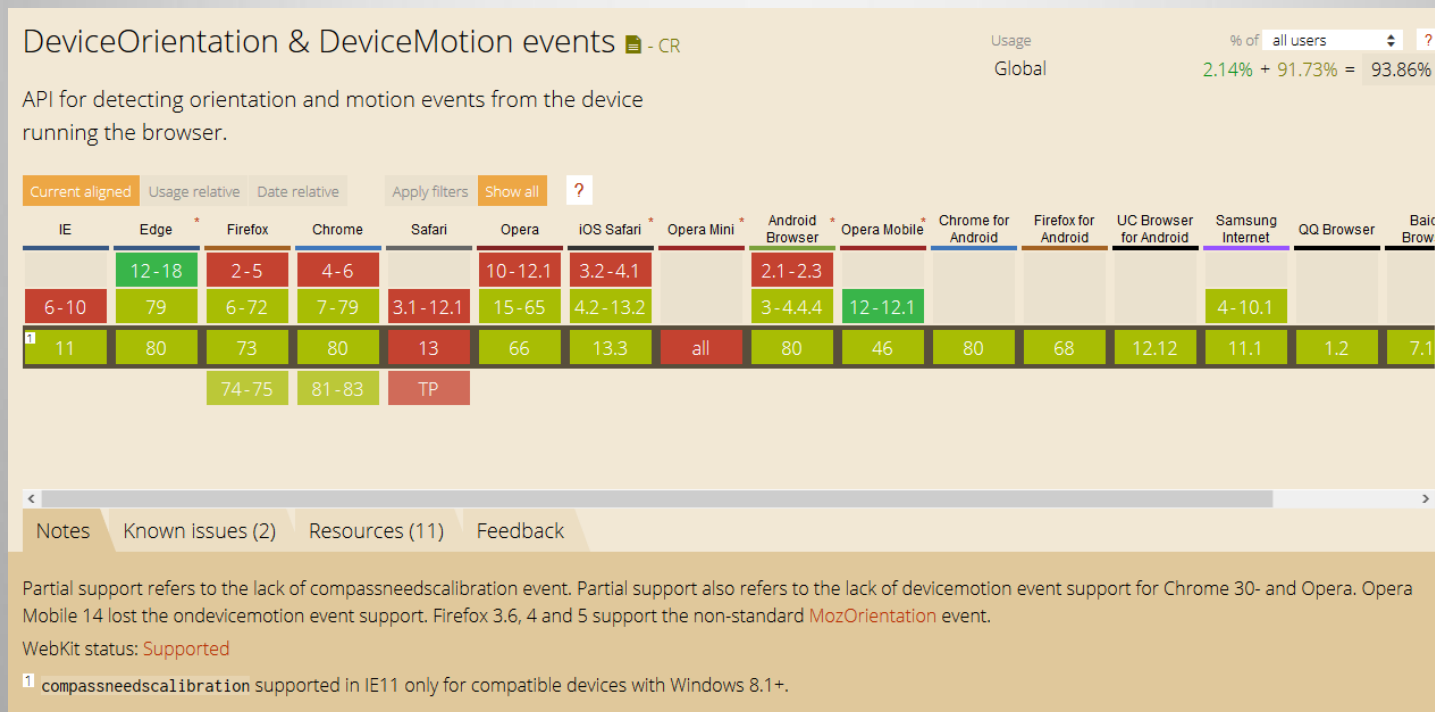


C'est donc un système à 10 capteurs d'attitude qui est embarqué

= 3 accéléromètres  
+ 3 gyromètres  
+ 3 magnétomètres  
+ 1 pression

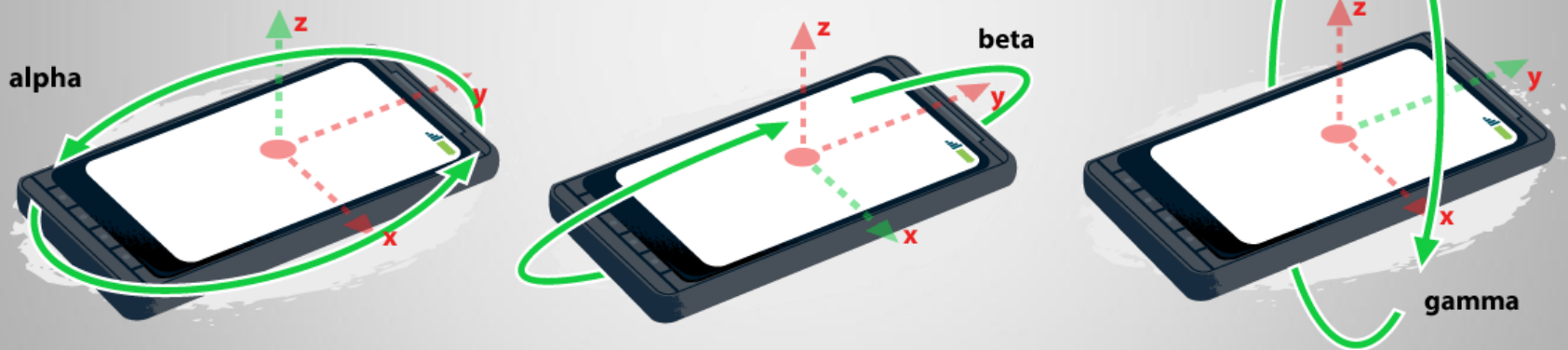
# DeviceOrientation Event Specification

- HTML5: attitude et mouvement du mobile
- <https://w3c.github.io/deviceorientation/>
- Canluse: DeviceOrientation et DeviceMotion 94%



# DeviceOrientation Event Specification

- DeviceOrientation:
  - Collecte les données d'inclinaison envoyées par l'accéléromètre
  - L'objet **event** retourne trois propriétés: alpha, beta, gamma

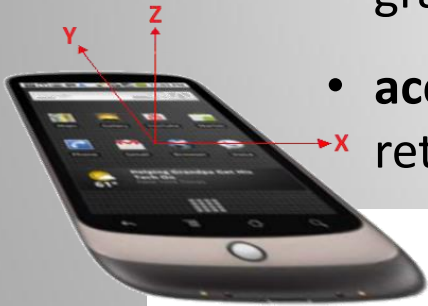


```
if(window.DeviceOrientationEvent) {  
    window.addEventListener("deviceorientation", process, false);  
} else {  
    // Le navigateur ne supporte pas l'événement deviceorientation  
}
```

JavaScript

# DeviceMotion Event Specification

- DeviceMotion:
  - collecte l'accélération sur les 3 axes ( $\text{m/s}^2$ )
  - L'objet **event** retourne deux propriétés :
    - **acceleration** : L'accélération calculée par l'appareil en enlevant la gravité.
    - **accelerationIncludingGravity** : La valeur de l'accélération brute, retournée par l'accéléromètre.



	Not accelerating	Accelerating up	Accelerating forward	Accelerating right	Accelerating up & to the right
<b>acceleration</b>	{0, 0, 0}	{0, 5, 0}	{0, 0, 2}	{3, 0, 0}	{5, 5, 0}
<b>accelerationIncludingGravity</b>	{0, 9.81, 0}	{0, 14.81, 0}	{0, 9.81, 2}	{3, 9.81, 0}	{5, 14.81, 0}

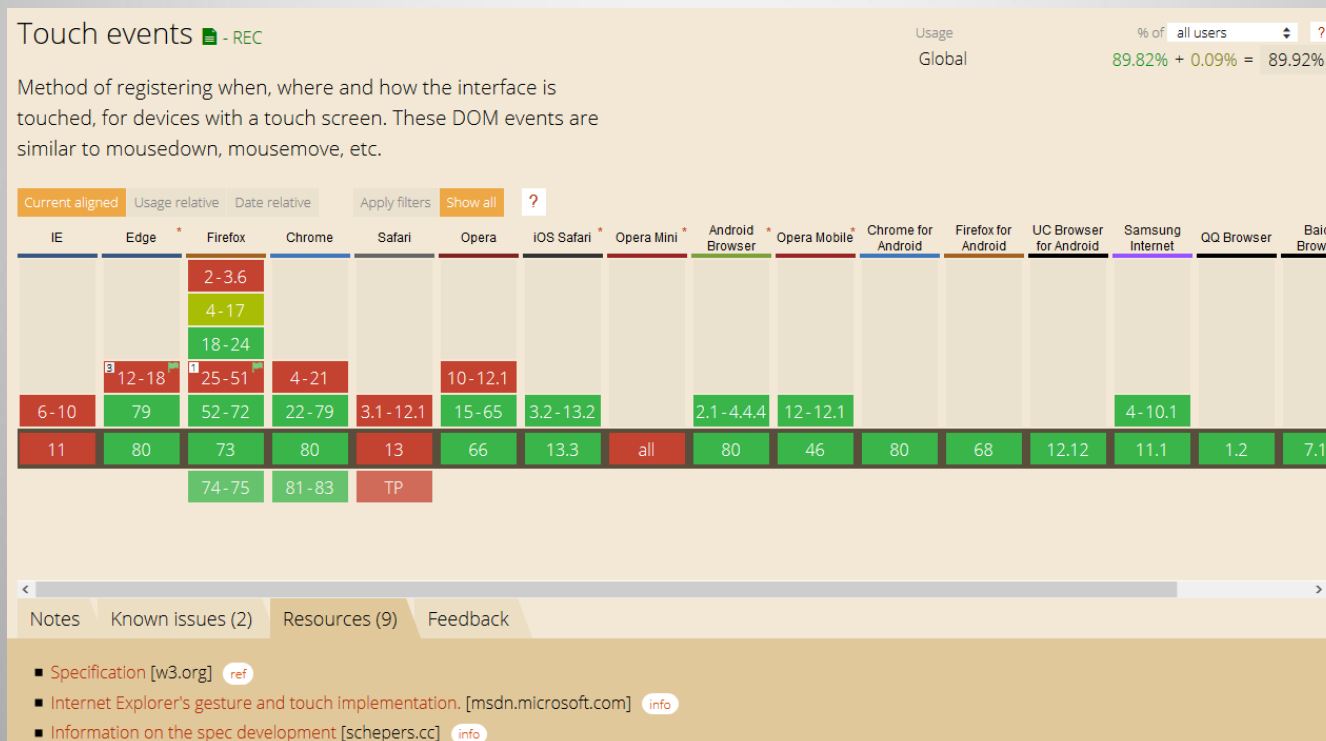
## Exercise 3

- Testez DeviceOrientation
- Testez DeviceMotion
- Sur votre smartphone
- <https://developers.google.com/web/tools/chrome-devtools/remote-debugging/>
- <https://developers.google.com/web/fundamentals/native-hardware/device-orientation/>



# Touch Event Specification

- HTML5: Events définis pour mouse/keyboard...mobile
- <https://www.w3.org/TR/touch-events/>
- Canluse: TouchEvents 90%



## Exercice 4

- **touchevents/mouse/...**
  - <https://dvcs.w3.org/hg/webevents/raw-file/tip/touchevents.html>
  - [https://www.w3schools.com/jsref/event\\_touchend.asp](https://www.w3schools.com/jsref/event_touchend.asp)
  - <https://developers.google.com/web/fundamentals/design-and-ux/input/touch/#touch-mouse-and-pointer-events>
- **Tester les touch events avec votre smartphone**

## Exercice 4 (bonus)

- **Combiner le tout:**
  - Utiliser Géolocalisation, DeviceOrientation et/ou DeviceMotion + touch events (bouton ou autre)
  - Lisser les données
  - 2 écrans: carte avec localisation et informations
  - Dessiner des rectangles de variable selon capteur
  - Dessiner un cercle ou flèche selon orientation
- **Envoie lien github page/code en fin de séance (ce soir maxi)** [Christophe.VESTRI@univ-cotedazur.fr](mailto:Christophe.VESTRI@univ-cotedazur.fr)