

# **Sensibilisation à la programmation multimedia**

<https://github.com/vestri/Initiation-multimedia>

**Christophe Vestri**

Le jeudi 18 septembre 2025

# Définition

- Qu'est-ce que le multimédia ?
- Définition. Technologie de l'information permettant **l'utilisation simultanée de plusieurs types de données numériques** (textuelles, visuelles et sonores) à l'intérieur d'une même application ou d'un même support, et cela, en y intégrant l'interactivité apportée par l'informatique.



Dans le cours:

- Géolocalisation
- Cartographie
- 3D
- IA



# Plan du cours

- 1<sup>er</sup> TD: Intro, github, carto/geo, leaflet/mapBox, rest Api
- 2em TD: 2D/3D: Canvas, WebGL et Three.js
- 3em TD: Three.js + Leaflet.js cartographie
- 4em TD: IA

## **Objectifs du cours:**

- Bases de géolocalisation et de la cartographie
- Initiation multimédia: 2D/3D, carto/géo et infographie
- Expérimenter quelques méthodes et outils web geo/3D
- Réaliser un petit projet (combinera ce qu'on a vu)

# Travail demandé

## Sur l'utilisation de l'IA et travail demandé:

- Commencez par lire les docs, comprendre avant de se lancer sur chatGPT.... Partez plutôt des docs pour apprendre
- Vous pouvez discuter, vous entraider mais le travail et le rendu sont individuels
- Le dernier TD sera full IA, à utiliser pour coder ou dans le projet (rendu toujours individuel)

## Evaluation:

- 25% par rendu de TD (et aussi participation au TD)
- N'oubliez pas de commiter, **pusher sur Github**
- **Vérifiez le fonctionnement sur Github et Smartphone**
- **Je corrige le soir ou le lendemain soir**

# Mon parcours

**Christophe Vestri**

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

DUT-Ingénieur-DEA-Thèse

The Airbus logo, consisting of the word "AIRBUS" in a bold, blue, sans-serif font.The IMRA logo, featuring a red circle with a white dot inside, followed by the letters "IMRA" in a bold, black, sans-serif font.

**3DVTech**

- Développement traitement image
- Bureau d'étude et conseil

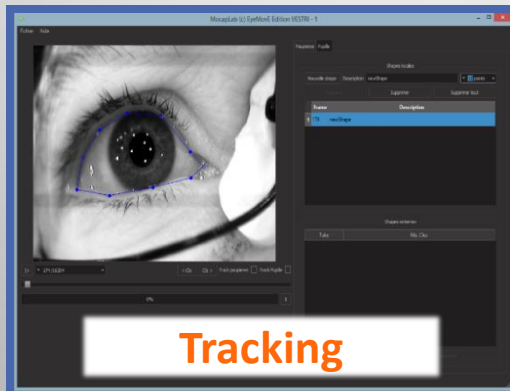
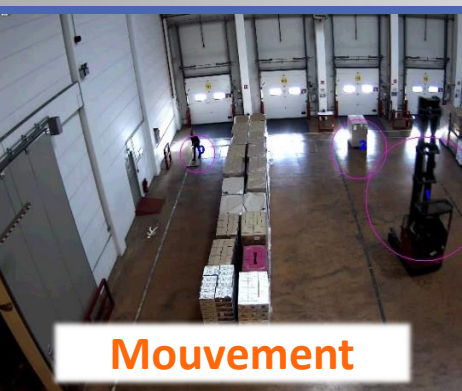
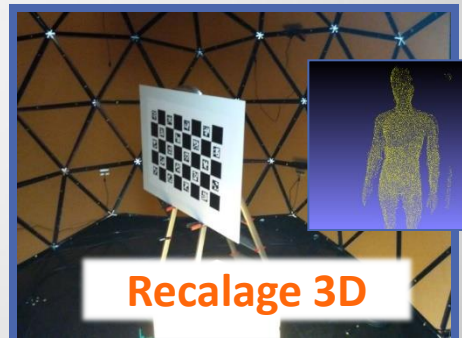
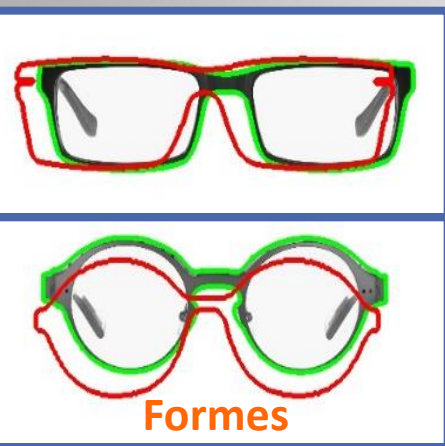
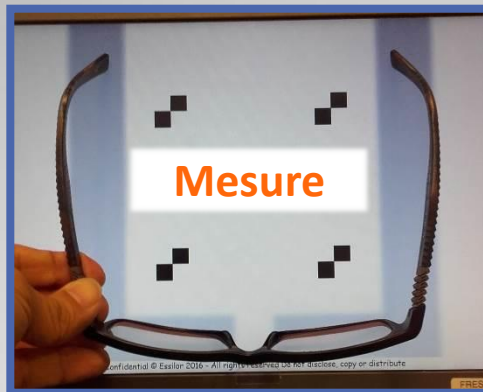
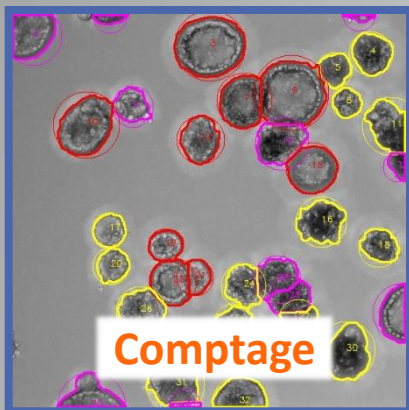
The 3DVTech logo, with "3D" in orange and "VTech" in blue, where the "V" is stylized with a blue outline.

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

**R&D Vision**

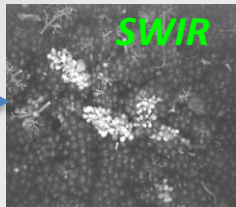
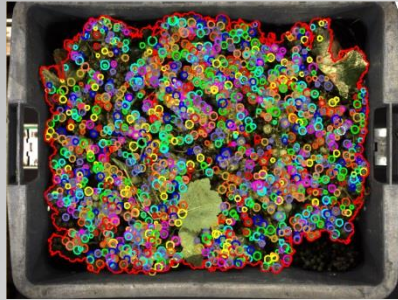
Expert traitement d'images

The R&D Vision logo, featuring three overlapping squares in blue, yellow, and purple on the left, and the text "R&D VISION" in a bold, black, sans-serif font on the right.





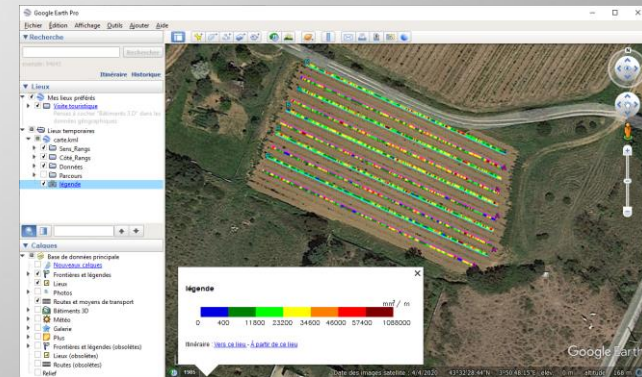
## Moët et Chandon / UPR



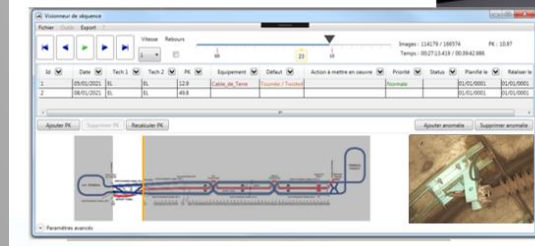
## Traitement du lin



## Gestion domaine agricole


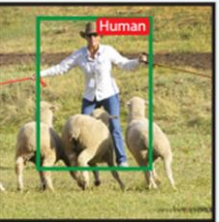
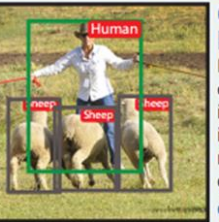





## Eurotunnel



# IA en Vision

## ► Exemples R&D Vision

 <p><b>Image Classification</b> Classify an image based on the dominant object inside it. <b>datasets:</b> MNIST, CIFAR, ImageNet</p>	 <p><b>Object Localization</b> Predict the image region that contains the dominant object. Then image classification can be used to recognize object in the region <b>datasets:</b> ImageNet</p>	 <p><b>Object Recognition / Detection</b> Localize and classify all objects appearing in the image. This task typically includes: proposing regions then classify the object inside them. <b>datasets:</b> PASCAL, COCO</p>
 <p><b>Semantic Segmentation</b> Label each pixel of an image by the object class that it belongs to, such as human, sheep, and grass in the example. <b>datasets:</b> PASCAL, COCO</p>	 <p><b>Instance Segmentation</b> Label each pixel of an image by the object class and object instance that it belongs to. <b>datasets:</b> PASCAL, COCO</p>	 <p><b>Keypoint Detection</b> Detect locations of a set of predefined keypoints of an object, such as keypoints in a human body, or a human face. <b>datasets:</b> COCO</p>

## ◆ AUTRES OUTILS

- ◆ OCR
- ◆ Tracking d'objets
- ◆ Segmentation automatique (Segment Anything Model)
- ◆ Profondeur à partir d'une image couleur

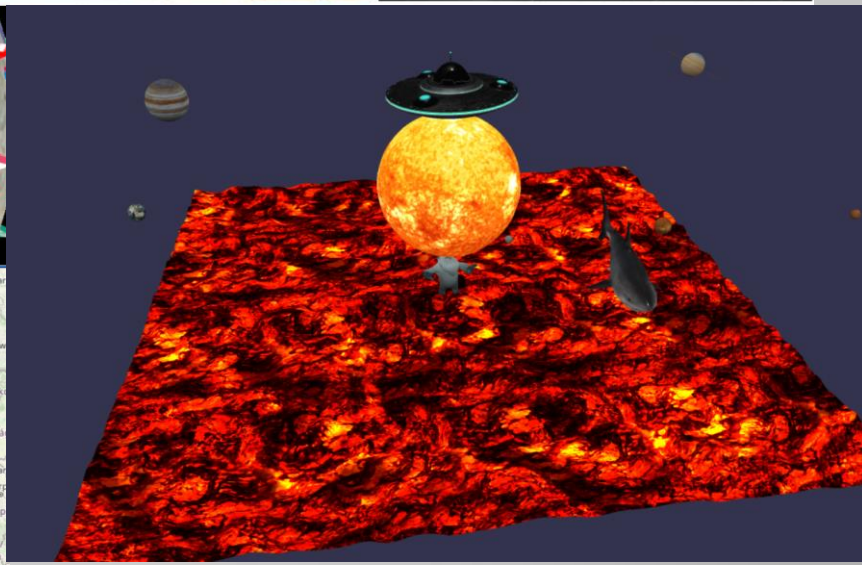
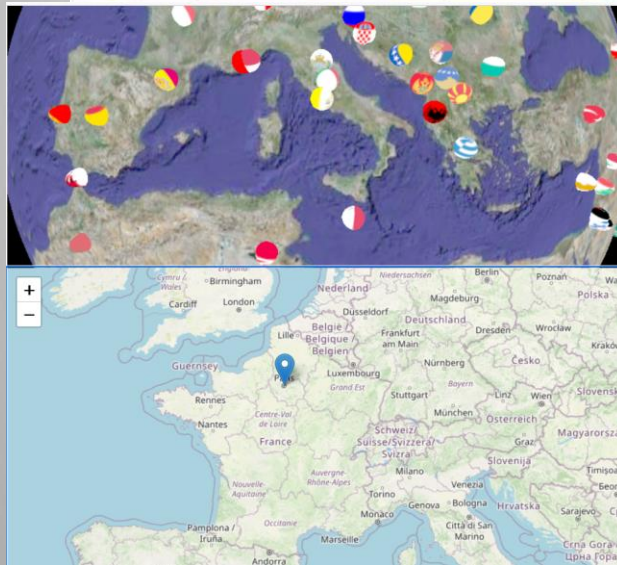


## Understanding Vision-Language Models





# Vous



# Plan Cours 1

- Git/Github/Github.io
- Géolocalisation, Capteurs smartphones
  - Geolocalisation HTML
  - Debug
- Référentiels et modèles de projection
- Leaflet

# Plan Cours 1

- **Git/Github/Github.io**
- Géolocalisation, Capteurs smartphones
  - Geolocalisation HTML
  - Debug
- Référentiels et modèles de projection
- Leaflet

# Pour tester sur un Mobile

## Git/Github/Github.io

- Créer un compte sur github
- Poster le code sur github.io ou une page
- Tester avec votre smartphone
- Récupérer le TD et les slides ici:  
<https://github.com/vestri/initiation-multimedia>
- Surtout: **Remplissez la google sheet**



# Git

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



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

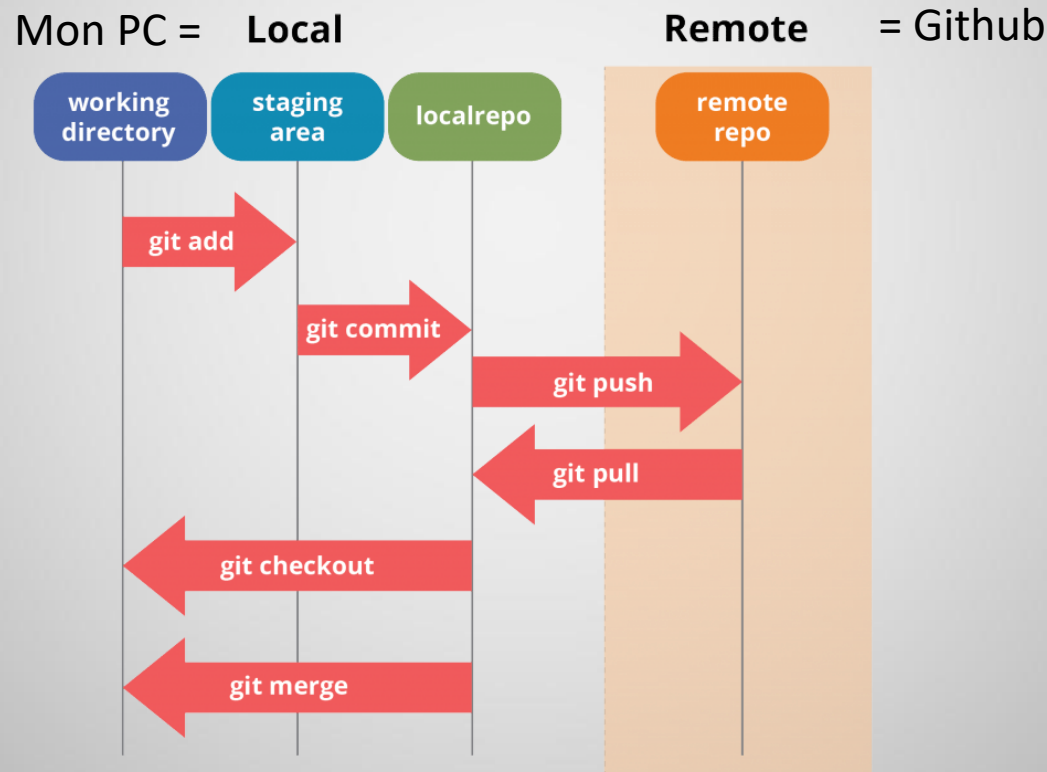
Git dispose notamment des commandes suivantes :

- `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

- Service d'hébergement «gratuit» utilisant git



# Page Github

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

# Premier travail

- Créer votre compte Github (si n'existe pas)
- Installer git ou tortoiseGit (git avec GUI)
- Installer une page pour le projet et les exercices:  
<https://docs.github.com/fr/pages/>
  - Créer projet, cloner le projet, ajoutez un code, 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 votrellogin.github.io -> [votrellogin.github.io](http://votrellogin.github.io)
- **Mettre les liens dans le googlesheet**

<https://github.com/vestri/initiation-multimedia>



# Plan Cours 1

- Git/Github/Github.io
- **Géolocalisation, Capteurs smartphones**
  - Geolocalisation HTML
  - Debug
- Référentiels et modèles de projection
- Leaflet

# Définitions

La **géolocalisation** 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

Techniques de [géolocalisation](#)

- Localisation par satellite (GNSS)
- 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

## 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 :

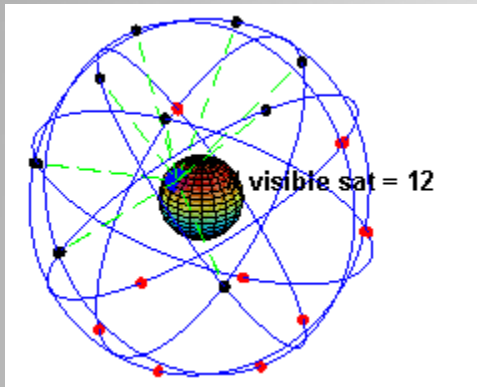
- [Beidou-1](#) pour la [Chine](#) ;
- [IRNSS](#) pour l'[Inde](#) (en cours de déploiement en 2015) ;
- [QZSS](#) pour le [Japon](#) (en cours de déploiement en 2015).

Comparaison des caractéristiques du segment spatial (2017)

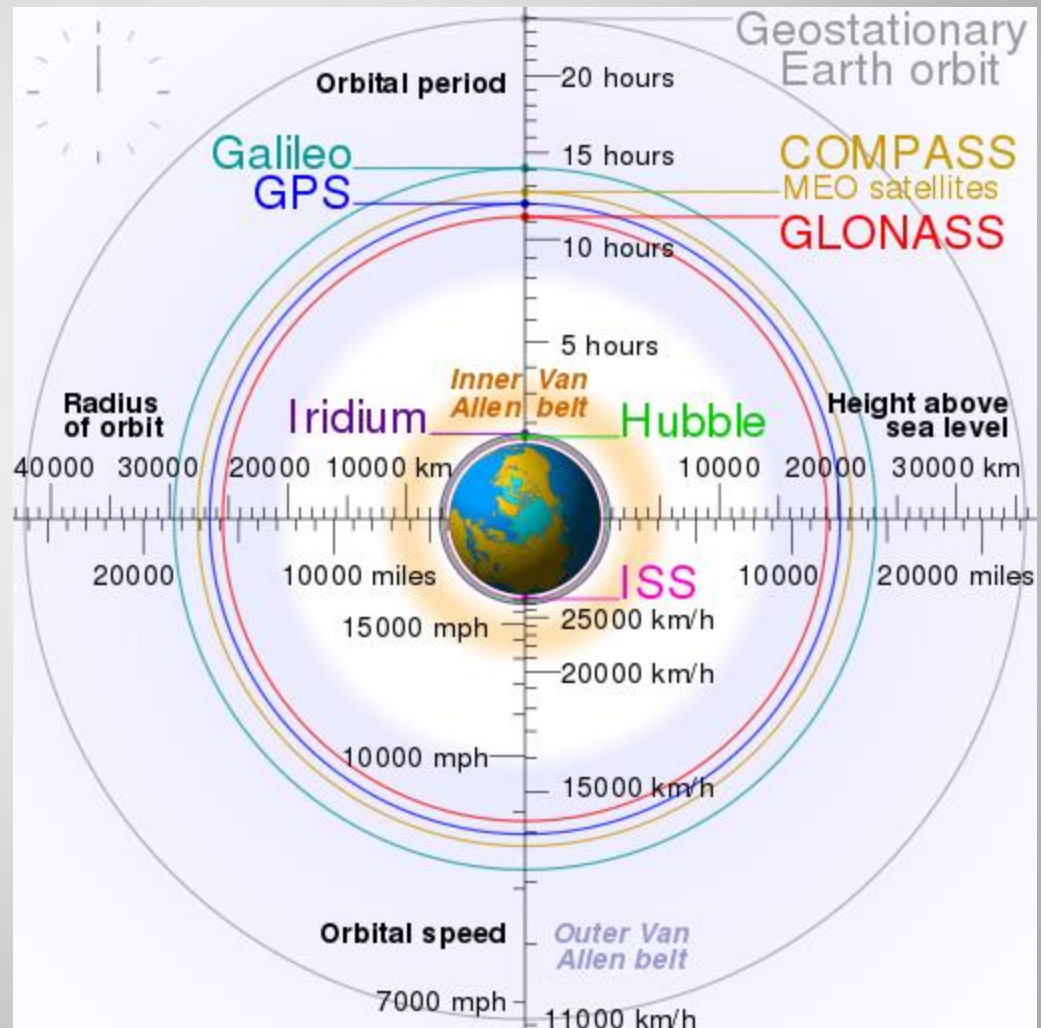
Caractéristique	GPS	GLONASS	GALILEO	Beidou/Compass
Segment spatial				
Altitude	20 200 km	19 100 km	23 222 km	21 528 km
Inclinaison	55°	64,8°	56°	55°
Période orbitale	11 h 58	11 h 15	14 h 07	12 h 53
Nombre de plans orbitaux	6	3	3	3
Nombre de satellites opérationnels (en cible)	31 (31)	24 (24)	15 (27)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



# Géolocalisation par Satellite



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



# Capteurs smartphones

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

# Applications



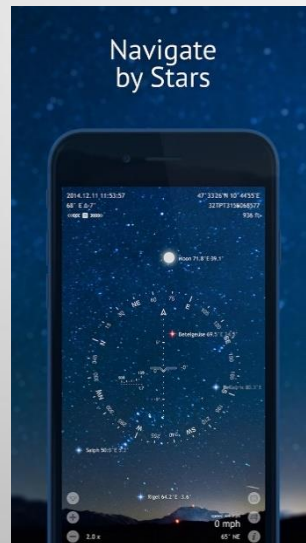
Pokemon Go



Immobilier



GPS



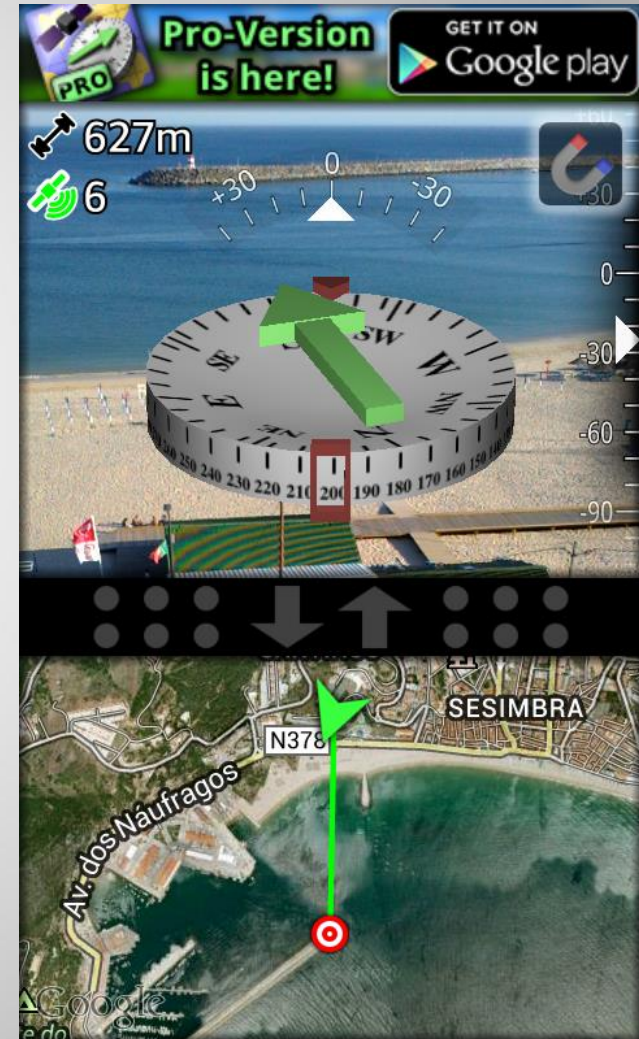
Recherche de points d'intérêts



# Ex avec géoloc+sensors

Utilisation des Capteurs du smartphone:

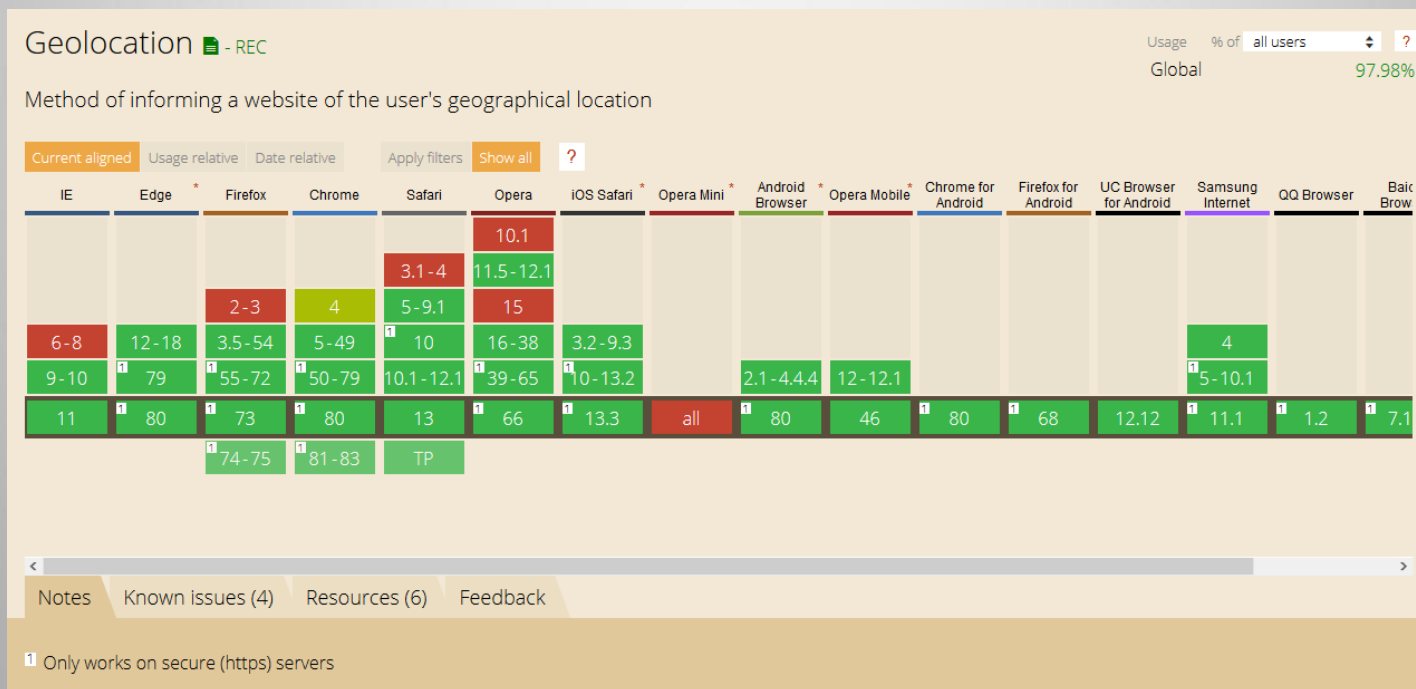
- GPS pour localiser son téléphone
- Recherche de Point d'intérêt proche de nous
- Mesure orientation (compas, accéléromètre)
- Augmente la réalité (RA)





# Geolocation Specification

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



# Outils de debug

- Page web en local:
  - `python3 -m http.server` (ou autre: wamp...)
  - <http://localhost:8000/> firefox ou chrome
- Debug:
  - Simple: F12 et suivre console (break points, erreurs)
  - Chrome: connecté à un smartphone:
    - <chrome://inspect/> (il faut tel android + option debug USB)
    - <https://developers.google.com/web/tools/chrome-devtools/javascript>
    - Simulation de smartphone, de localisation, etc... (F12)
- Firefox possible ou autres??

# Exercice 1

- Testez accès Geolocalisation
- Afficher
  - sa position lon, lat, hauteur
  - la précision de mesure
  - sa vitesse
  - Le time stamp
- Tester avec/sans gps
- utilisez `getCurrentPosition()` et `watchPosition()`
- Testez le debug (en local=F12 ou sur github=chrome inspect)

# Plan Cours 1

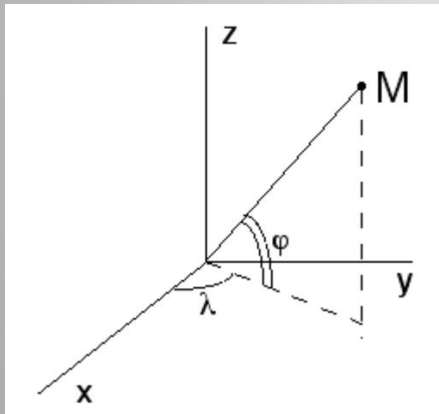
- Git/Github/Github.io
- Géolocalisation, Capteurs smartphones,
  - Geolocalisation HTML
  - Debug
- **Référentiels et modèles de projection**
- Leaflet

# Systèmes Géographiques et Cartographiques

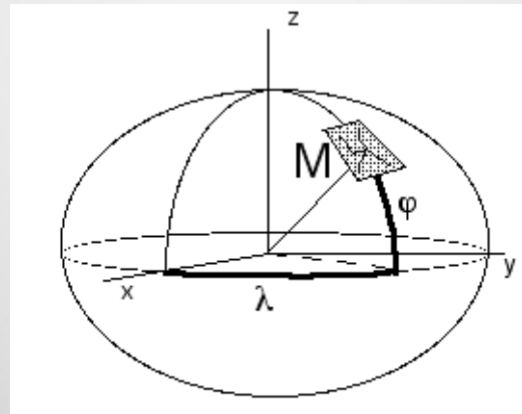
- Construction d'un référentiel géographique

Choix d'un ellipsoïde

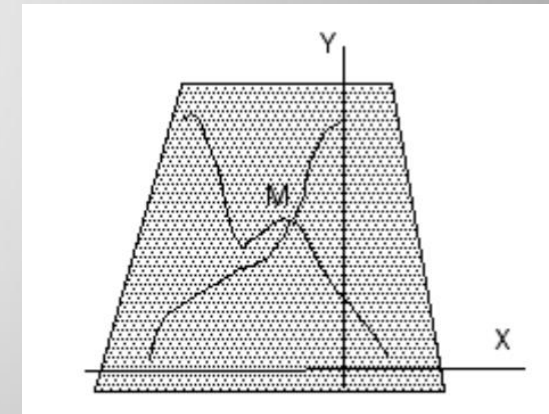
Choix d'une projection



Système cartésien  
 $x, y, z$   
(ex: dans la pièce)



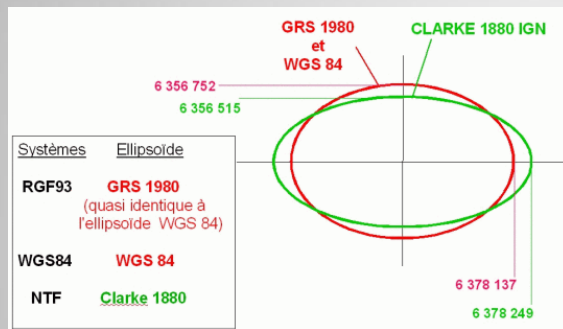
Système géographique  
 $\varphi, \lambda, Z$   
(gnss/gps)



Système cartographique  
 $X, Y, Z$   
(ex: cartes IGN, googlemaps)

# Systemes Géographiques et Cartographiques

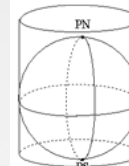
- Construction d'un référentiel géographique



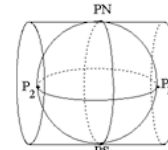
Choix d'un ellipsoïde

Représentation cylindrique :

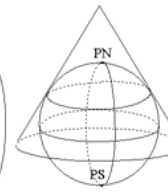
Représentation conique :



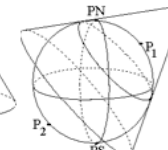
directe



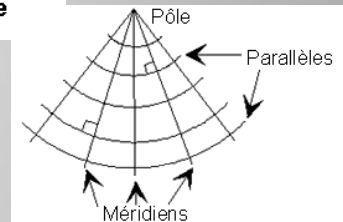
transverse



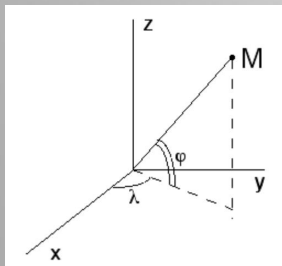
directe



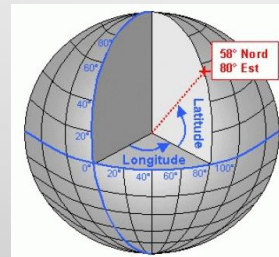
oblique



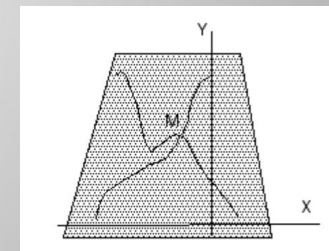
Choix d'une projection



Système cartésien  
 $x, y, z$



Système géographique  
 $\varphi, \lambda, z$

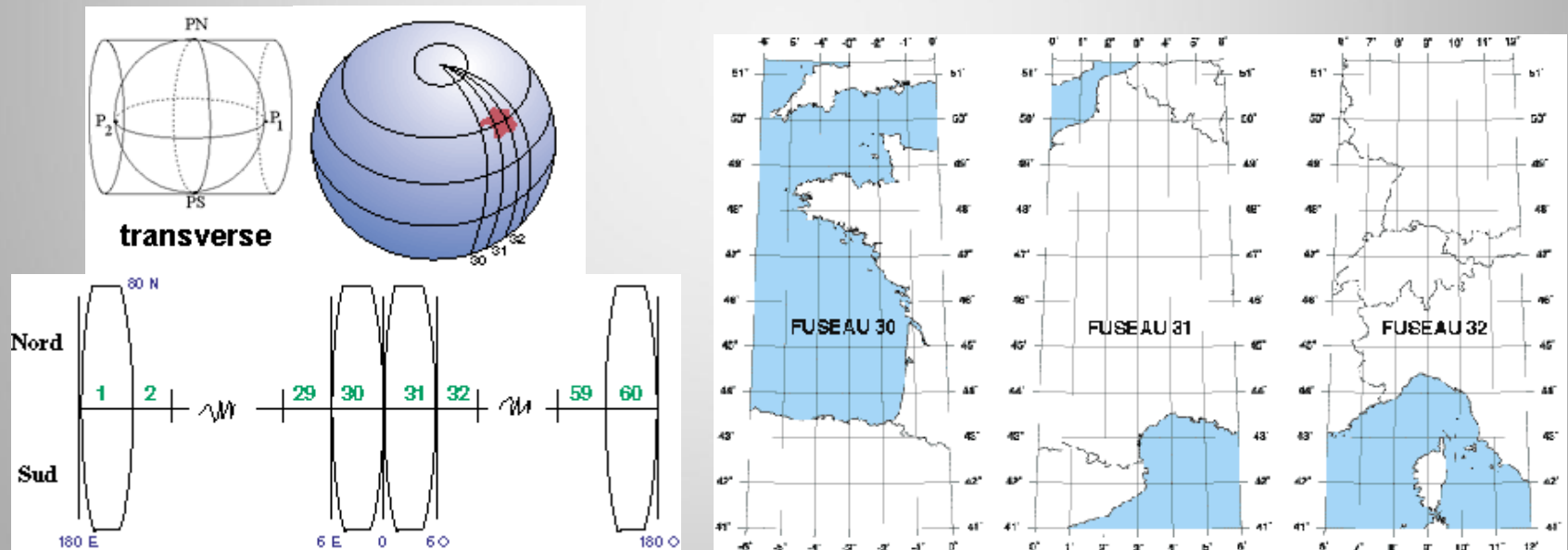


Système cartographique  
 $X, Y, z$



# Systèmes Géographiques et Cartographiques

- GPS: UTM (Universal Transverse Mercator)
  - Système mondial de 122 projections
  - 60 **fuseaux** de  $6^\circ$  (entre  $80^\circ$  Sud et  $80^\circ$  Nord) + 2 poles



- La France: fuseaux UTM Nord 30, 31 et 32

# Systèmes Géographiques et Cartographiques

- Coordonnées GPS: Lat/Lon
  - La salle 202:  
 $43.616513, 7.072094 = 43^{\circ}36'59.5''\text{N} + 7^{\circ}04'19.5''\text{E}$
- 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\\_cooronnees.htm](http://sgcaf.free.fr/pages/techniques/ign_cooronnees.htm)

# Leafletjs

- [leafletjs](https://leafletjs.com/) est une librairie Opensource pour afficher des cartes interactives utiles à la navigation (comme google maps)
- Seulement 33Ko, Tous les browsers
  - Map controls
  - Layers
  - Interaction Features
  - Custom maps



## Exercice 2

- Avec Leafletjs (chaque fonction 1 ou 2pt)
  - Récupérez votre position GPS (Exo1), afficher votre position sur la carte
  - Afficher une carte locale (utilisez openStreetmap)
  - Affichez un marqueur sur Nice
  - Tracez le triangle des Bermudes (en rouge)

**Testez en local puis publiez sur Github**

# Exercice 2 suite

- Avec Leafletjs ou mapbox (faire un max des propositions)
  - Changer de carte (par ex: stamen: <http://maps.stamen.com/>)
  - Dessiner un cercle autour de votre position avec une rayon représentant la précision estimée
  - Calculez la distance à Marseille, l'afficher ([https://fr.wikipedia.org/wiki/Distance\\_du\\_grand\\_cercle](https://fr.wikipedia.org/wiki/Distance_du_grand_cercle))
  - Récupérer des données géoréférencées et les afficher sur la carte
    - Fichier geoJson local, Requête html
    - requête API (<https://api.gouv.fr/documentation/api-geo> )
  - afficher un trajet/route (google/mapbox/mapQuest)
    - Testez d'autres outils
      - OSRM: <https://project-osrm.org/>
      - [mapBox](#), [google maps api](#)

# Liens

- Geojson sur <http://opendata.nicecotedazur.org>
- ou par une autreApi :
  - <https://adresse.data.gouv.fr/api>
  - <https://api.gouv.fr/api/api-geo.html>
  - <https://www.insee.fr/fr/metadonnees/cog/departement/DEP06-alpes-maritimes>
  - <https://www.data.gouv.fr/fr/>
- Idéalement, requête, sinon fichier uploadé sur gitub



# Liens

- Requete HTML
  - [https://developer.mozilla.org/fr/docs/Web/API/XMLHttpRequest/Using\\_XMLHttpRequest](https://developer.mozilla.org/fr/docs/Web/API/XMLHttpRequest/Using_XMLHttpRequest)
  - <https://leafletjs.com/examples/geojson/>
- Exemple avec API Geo
  - <https://api.gouv.fr/documentation/api-geo>
  - <https://geo.api.gouv.fr/communes?codePostal=06330&fields=nom,code,codePostaux,codeDepartement,codeRegion,population&format=json&geometry=centre>
  - La reponse est un fichier GeoJson

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

## Response body

```
[
  {
    "nom": "Roquefort-les-Pins",
    "code": "06105",
    "codePostaux": [
      "06330"
    ],
    "codeDepartement": "06",
    "codeRegion": "93",
    "population": 6695
  }
]
```

# With Leaflet

[https://ckan.publishing.service.gov.uk/dataset/?license\\_id=uk-ogl&\\_tags\\_limit=0&tags=Outline&res\\_format=GeoJSON](https://ckan.publishing.service.gov.uk/dataset/?license_id=uk-ogl&_tags_limit=0&tags=Outline&res_format=GeoJSON)

```
let xhr = new XMLHttpRequest();
xhr.open('GET', 'uk_outline.geojson');
xhr.setRequestHeader('Content-Type', 'application/json')
  xhr.responseType = 'json';
xhr.onload = function() { if (xhr.status !== 200) return
L.geoJSON(xhr.response).addTo(map); };
xhr.send();
```

Ou