Sensibilisation à la programmation multimedia

Christophe Vestri

Le mercredi 15 octobre 2025

Plan du cours

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

Objectifs du cours:

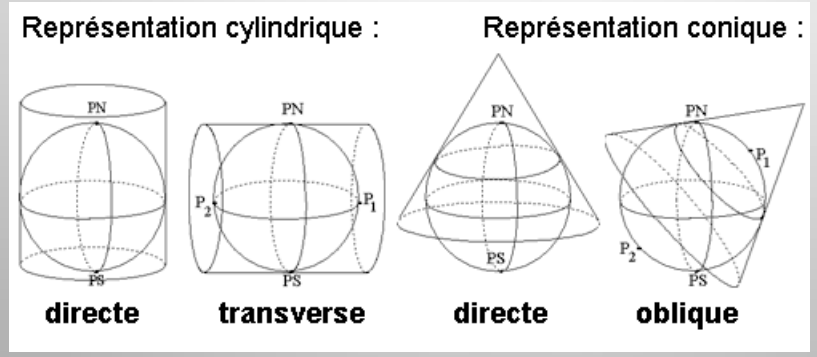
- Bases de géolocalisation et de la cartographie
- Initiation multimédia: 2D/3D, carto/géo et infographie
- Expérimenter quelques méthodes et outils web geo/3D
- Réaliser un petit projet (combinera ce qu'on a vu)

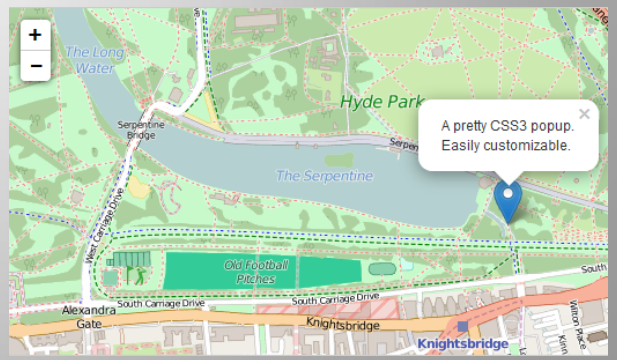
Plan Cours 4

- Rappel dernier cours + Questions
- Comment programmer avec une IA
- Exercice IA:
  - Qqs propositions de sujet
  - Projet libre si tourne autour de:
    - Géolocalisation et cartes
    - 3D + orientation smartphone

Géo + Html5 + LeafLet.js

- Repères Géo et carto
- Acces capteur caméra: Géolocalisation, DeviceOrientation, DeviceMotion
- Leafletjs, Mapbox, mapQuest
- Données géolocalisées (REST API)



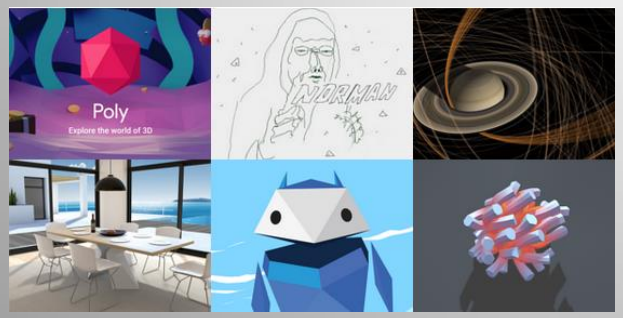


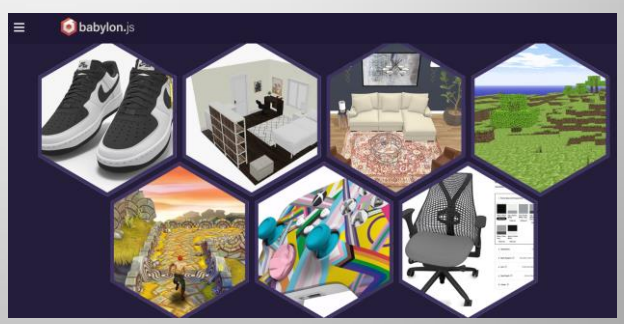
3D Three.js et Babylon.js

- **3D sur le web** THREE.JS babylon.js
  - **Libraries 3D dehaut niveau de WebGL**
  - **Cross-plateforme et gratuit**

Scenes, Cameras, Renderer, Geometry, Materials, Textures, Lights, loaders

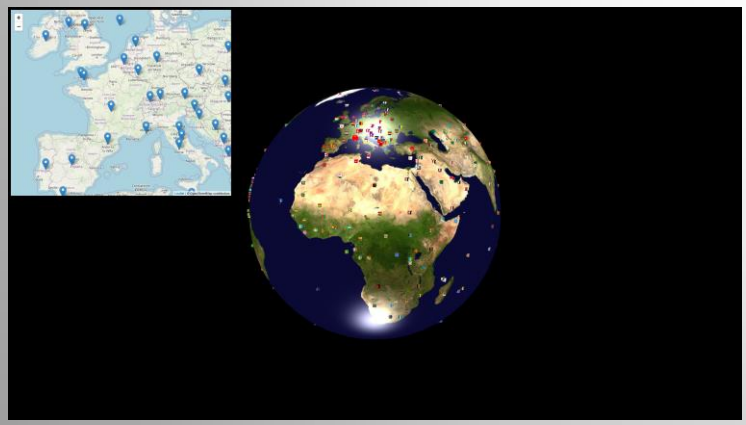
  - **https://threejs.org/**
  - **https://www.babylonjs.com/**

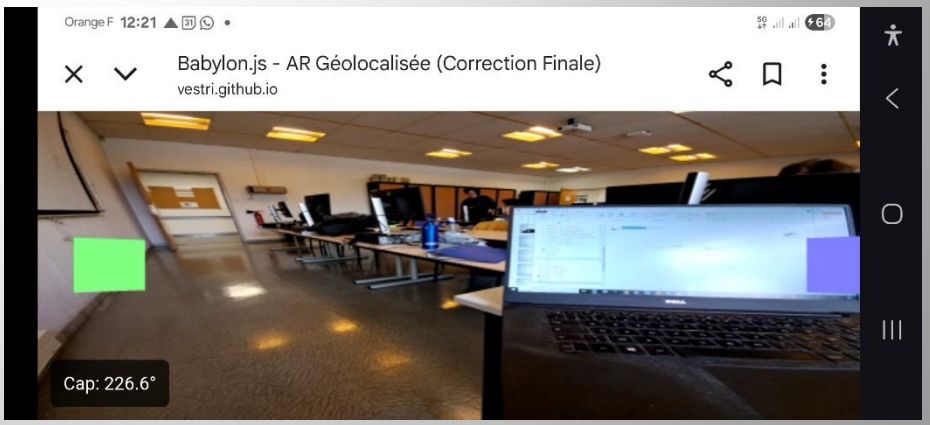




3D+2D et location-based RA

- **Données géolocalisées**
  - **2D et 3D**
- **Location-based RA**





Comment programmer avec une IA

- Différents outils
  - Github Copilot, Cursor, Claude IA, Gemini, GPT
  - Intégré dans des IDE ou non:Visual code, visual studio…
- Fonctionnement
  - Avant: prompt -> réseau de neurone -> réponse
  - Actuel: txt+img+… -> agents + instructions +web -> résponse
- 3 principaux modes de fonctionnement
  - Prompting
  - Instruction
  - MCP

Prompting

- Le plus simple:
  - On donne des instructions, il propose une réponse comme tous les LLM
- Les problèmes récurrents
  - Fonctions erronées: prompt trop vague
  - Hallucination: utilise des fonctions qui n’existent pas
  - Code non fonctionnel: ne compile pas, aucun test
  - Code obsolète: anciennes versions de librairies

Prompting

- Comment faire 1 bon prompt?
  - Contexte: expliquer le contexte, son rôle
    - Code C++, librairie OpenCV, traitement d'image expert
  - Besoin: exprimer clairement ses attentes
    - Fonction qui fait ca, 4 étapes, (1) vérifie les données, (2)….
  - Détail: donner un maximum de détail sur les intérecations ou ce que l'on veut dedans
    - 4 étapes, (1) vérifie les données, (2)….
    - La sortie doit être comme ci, en argument une fonction qui …
    - Exemple de code ou de chaine de pensée
  - Format:
    - La sortie: 1 pdf, 1 json, un texte de 3 lignes, une image svg….
- https://www.promptingguide.ai/techniques

Instructions

- Prompting=On repete souvent la même chose
- Solution instructions
  - Liste de consignes ou d'étape à appliquer dans son raisonnement
  - Taches répétitives: simplification de code, documentation, commit…
  - Global ou local, dans des fichiers
- Exemple:
  - Guideline pour coder
  - Guideline pour documenter du code

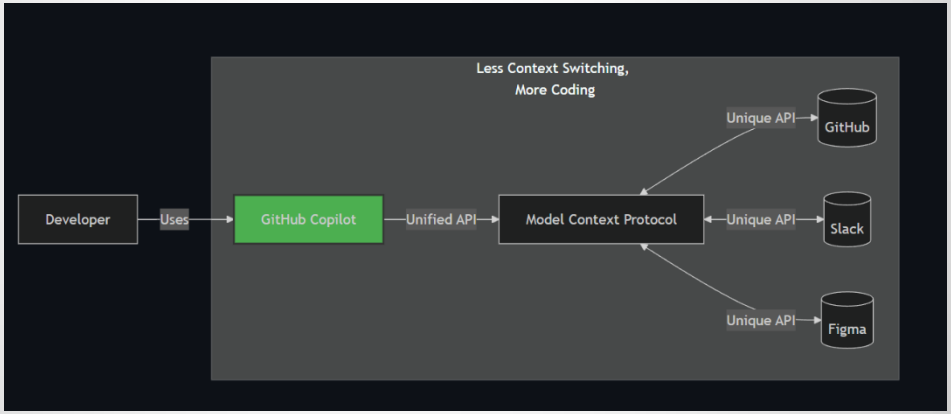
Instructions

```
# Clean Code Guidelines

## Constants Over Magic Numbers
- Replace hard-coded values with named constants
- Use descriptive constant names that explain the value's purpose
- Keep constants at the top of the file or in a dedicated constants file

## Meaningful Names
- Variables, functions, and classes should reveal their purpose
- Names should explain why something exists and how it's used
- Avoid abbreviations unless they're universally understood

## Smart Comments
- Don't comment on what the code does - make the code self-documenting
- Use comments to explain why something is done a certain way
- Document APIs, complex algorithms, and non-obvious side effects

## Single Responsibility
- Each function should do exactly one thing
- Functions should be small and focused
- If a function needs a comment to explain what it does, it should be split

## DRY (Don't Repeat Yourself)
- Extract repeated code into reusable functions
- Share common logic through proper abstraction
- Maintain single sources of truth

## Clean Structure
- Keep related code together
- Organize code in a logical hierarchy
- Use consistent file and folder naming conventions

## Encapsulation
- Hide implementation details
- Expose clear interfaces
- Move nested conditionals into well-named functions

## Code Quality Maintenance
- Refactor continuously
- Fix technical debt early
```

```
## Goals

- Document existing project structure and tech stack.
- Ensure established practices are followed.
- Minimize bash command and build failures.

## Limitations
- Instructions must be no longer than 2 pages.
- Instructions should be broadly applicable to the entire project.

## Guidance

Ensure you include the following:

- A summary of what the app does.
- The tech stack in use
- Coding guidelines
- Project structure
- Existing tools and resources

## Steps to follow

- Perform a comprehensive inventory of the codebase. Search for and view:
  - README.md, CONTRIBUTING.md, and all other documentation files.
  - Search the codebase for indications of workarounds like 'HACK', 'TODO', etc.
- All scripts, particularly those pertaining to build and repo or environment setup.
- All project files.
- All configuration and linting files.
- Document any other steps or information that the agent can use to reduce time spent exploring or trying and failing to run bash commands.

## Validation

Use the newly created instructions file to implement a sample feature. Use the learnings from any failures or errors in building the new feature to further refine the instructions file.
```

MCP

- Model Context Protocol
- Permet de communiquer avec des librairies/outils uptodate



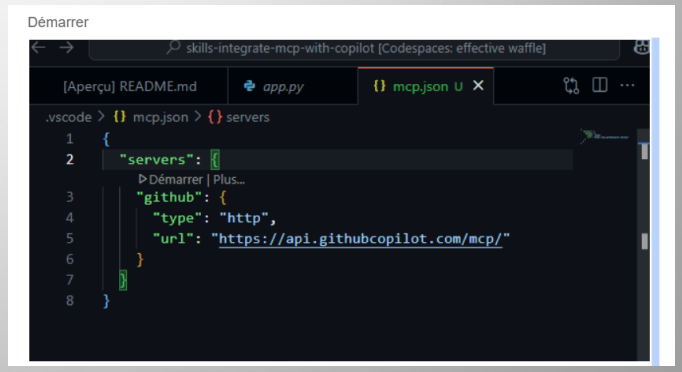
- Exemple sous visual code

Ajout d'un MCP

Aller dans le dossier .vscode
Ajouter un fichier mcp.json

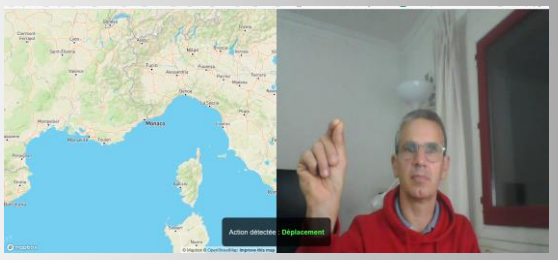
Et par exemple ajouter le lien vers le mcp de github

```
{
  "servers": {
    "github": {
      "type": "http",
      "url": "https://api.githubcopilot.com/mcp/"
    }
  }
}
```



Exercices d'IA

- **Objectif**= pratiquer ou utiliser des IA pour faire
  - Une application fonctionnelle
  - Pas ou peu de codage, faite le débuguer ou corriger ces erreurs
  - C'est vous le guide, vous devez maitriser ce qu'il fait
    - (1) aller pas à pas ou (2) donnez beaucoup de détails



- **Qqs propositions de sujets**
  - **Carte interactive IA**: ajoutez à vos applications un contrôle par IA (ml5.js)
  - Refaire le **TD3 en 100% IA**
  - **Boussole 3D intelligente**: caméra + compa 3D en 100% IA
  - **Reconnaissance de lieu 3D** (tensor.js/ml5.js), flux caméra, détection
    - Ex: tour eiffel -> afficher Modèle 3D, photo 2D ou lien page wikipédia…
  - **Projet libre** si tourne autour de:
    - Géolocalisation et cartes
    - 3D + orientation smartphone

Fin du cours

N’oubliez pas de commiter, pusher sur Github
Vérifiez le fonctionnement

bravo à tous