INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES STRASBOURG

AUTODESK
Make Anything

Mettre en place un plugin Civil 3D / AutoCAD avec l'IA

Notice d'installation, utilisation et bonnes pratiques

Rédacteur : PELISSIER Jean
Relecture : BERSON Guillaume

Janvier 2026

Pourquoi développer un plugin ?

Automatiser des tâches répétitives

Réduire le temps passé sur des actions manuelles répétitives et limiter les erreurs humaines en automatisant vos opérations longues et sensibles.

Outils adaptés à vos besoins

Créer des fonctionnalités parfaitement adaptées à vos besoins métiers et propres à vos méthodes de travail, afin d'intégrer vos calculs, contrôles et méthodes internes directement dans Civil 3D.

Méthodes unifiées

Garantir une méthode de travail identique pour tous, avec des paramètres choisis pour votre entreprise et des résultats reproductibles sur l'ensemble de vos projets.

Créer de la valeur interne

Développer des plugins permet de créer des outils durables qui renforcent l'expertise de l'équipe, soutiennent l'innovation et modernisent vos méthodes de production.

Comment crée-t-on un plugin Civil 3D ?

Autodesk met à disposition l'API complète de Civil 3D, accessible en C# via .NET.

Un développeur crée un projet .NET, programme les commandes nécessaires, compile une DLL et la charge dans Civil 3D via NETLOAD.

Ce processus demande du temps, une bonne connaissance de l'API et de la logique interne du logiciel.

Qu'est-ce que Cursor ?

Cursor est un environnement de développement (IDE) basé sur l'intelligence artificielle, pensé pour écrire du code avec vous.

Contrairement à un simple chatbot comme ChatGPT ou Gemini, Cursor a accès à vos fichiers, peut analyser vos projets, comprendre le contexte et créer ou modifier du code directement dans un éditeur.

L'expérience est fluide et intuitive : c'est du "vibe coding".

Suite à une discussion avec l'IA, elle peut :

- Créer un environnement pour votre plugin
- Coder en C# en utilisant l'API de Civil3D
- Compiler le code généré et build le DLL du plugin dans son terminal
- Corriger des erreurs dans un code jusqu'au bon fonctionnement de celui-ci.

Alternatives à Cursor

VSCode + Github Copilot

Rider + JetBrain AI

VSCode + Codex

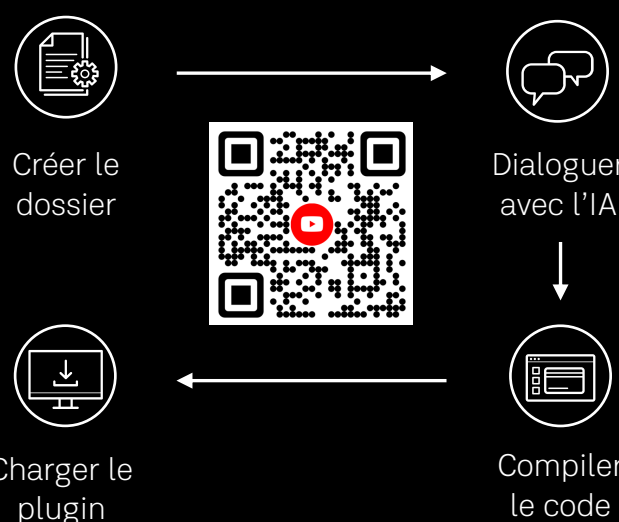
Pré-requis pour développer avec Cursor

Pour développer un plugin Civil 3D avec Cursor, il faut :

Une version Windows Civil3D/Autocad

.NET SDK installé pour compiler

Workflow simple



Bonnes pratiques

L'utilisation de Cursor n'est pas « magique », il existe des méthodes et des bonnes pratiques à suivre :

- Utiliser un prompt optimisé : La qualité de votre prompt détermine la qualité du plugin. Il est primordial de suivre les étapes décrites dans le document de rédaction de prompt. Cela passe par une méthodologie d'écriture précise. Vous pouvez d'ailleurs utiliser une autre IA pour rédiger ce prompt.
- Tester et faire des retours : Il est possible que le plugin ne soit pas fonctionnel du premier coup. Il est important de le tester dans Civil 3D et de retourner le message d'erreur, ou de préciser ce que vous souhaitez ajouter ou modifier.
- Avoir une structure de projet claire : Cursor est plus performant dans un environnement de projet organisé, avec des dossiers et sous-dossiers bien structurés.
- Configurer Cursor en mode semi-automatique : Il est possible d'activer une option semi-auto dans Cursor qui l'oblige à vous faire valider chaque étape manuellement, afin que vous gardiez un contrôle complet sur le développement du plugin.

Erreurs à éviter

Certaines erreurs qui peuvent paraître anodines ont tendance à faire entrer l'IA dans des boucles d'erreurs :

- Un "wording" imprécis ou mal choisi : Cursor suit la logique que vous décrivez. Un terme inadapté peut l'envoyer vers une mauvaise solution (ex : polyligne au lieu d'axe) et provoquer des erreurs en boucle. La précision du vocabulaire est essentielle.
- Penser que toutes les fonctions visibles dans Civil 3D existent dans l'API : Certaines commandes ou options disponibles dans l'interface peuvent ne pas exister dans l'API fournie par Autodesk. Cursor peut donc chercher des méthodes qui n'existent pas. Il faut garder à l'esprit que UI ≠ API.
- Oublier de donner à Cursor les éléments techniques indispensables :
    - Chemins vers les DLL Autodesk
    - La version de Civil 3D utilisée
    - La version du .NET SDK
    - La structure du projet

  Un plugin ne se décrit pas seulement par son objectif, il faut décrire l'environnement technique.

Conclusion

Créer des plugins Civil 3D peut donc être une bonne manière d'améliorer vos workflows, automatiser des tâches répétitives et gagner en efficacité sur vos projets. Cursor peut générer ces plugins de manière rapide et intuitive, à condition de s'appuyer sur une bonne connaissance métier et un prompt optimisé dont vous retrouverez l'explication dans la section suivante.