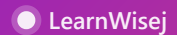


Exercice de code avec l'IA

Construisez l'exemple VisualOperationsStudio de ce module avec ChatGPT ou Claude à partir d'un prompt prêt à l'emploi qui oriente le modèle vers la spécification du module, puis relisez, exécutez et étendez ce qu'il vous renvoie.



L'assistant, ce sont les mains rapides. L'architecte, c'est toujours vous.

Un assistant peut produire l'exemple de ce module en quelques minutes. Ce qu'il ne peut pas faire, c'est décider si le résultat est la bonne conception pour votre application — cette décision vous revient. C'est vous qui apportez le jugement : briefez-le correctement, puis vérifiez ce qu'il vous renvoie à la lumière de ce que le Module 2 vient de vous apprendre.

Ce que vous demandez

Ajoutez à VisualOperationsStudio un contrôle TelemetryGauge réutilisable qui dessine un anneau, des zones de seuil colorées, une aiguille, une valeur numérique et une légende. Dotez-le de propriétés Value, Minimum, Maximum, WarningThreshold et CriticalThreshold dont les setters bornent la valeur reçue, sortent immédiatement si rien n'a changé et appellent Invalidate() une seule fois. Placez les calculs de coordonnées dans un helper pur GaugeGeometry qui prend un Rectangle et renvoie les limites de l'anneau, les angles de balayage, les points de l'aiguille et le rectangle du libellé, puis dessinez à partir de ces valeurs avec des pens, des brushes et un GraphicsPath portés par des using, sans toucher au Graphics fourni. Gérez sans lever d'exception un contrôle de taille nulle, une plage inversée et une valeur hors plage, et exposez la mesure via AccessibleDescription pour qu'elle n'existe pas uniquement sous forme de pixels.

Le module demande ces livrables, et le prompt ci-dessous demande exactement ceux-là à l'assistant :

- ✓ Contrôle TelemetryGauge avec des propriétés Value, Minimum, Maximum et de seuil qui bornent la valeur et appellent Invalidate() une fois par changement réel
- ✓ Helper pur GaugeGeometry renvoyant les limites de l'anneau, les angles de balayage, les points de l'aiguille et le rectangle du libellé
- ✓ Gestionnaire Paint qui libère chaque pen, brush, font et path qu'il crée et ne libère jamais e.Graphics
- ✓ Garde-fous pour une taille nulle, une plage inversée et des valeurs hors plage, avec une AccessibleDescription portant la mesure
- ✓ PaintCost.md indiquant la taille du redessin, sa fréquence et les cas où un contrôle théme coûterait moins cher

Pourquoi il est conçu ainsi

Le Module 1 a placé une même mesure de télémétrie sur quatre surfaces pour vous montrer où chaque pixel est produit. VisualOperationsStudio veut désormais cette mesure à six endroits, et recopier un gestionnaire Paint dans six panneaux, c'est ainsi qu'un écran se retrouve avec six jauges légèrement différentes, six pens jamais libérés et un redessin dès qu'un champ sans rapport change. La solution est d'arrêter d'écrire des gestionnaires et d'écrire un contrôle: un TelemetryGauge unique dont les propriétés publiques sont la seule chose capable de changer l'image, et dont vous pouvez énoncer le coût en kilo-octets.

Voici les termes que l'assistant doit maîtriser. Si ce qu'il vous renvoie les emploie de façon approximative, il n'a pas compris le module :

Terme	Ce qu'il signifie ici
Wisej.Web.PaintEventArgs	L'argument de l'événement Paint; il vous remet un System.Drawing.Graphics que vous n'avez pas créé et un rectangle à l'intérieur duquel dessiner, et votre code s'exécute sur le serveur.
ClipRectangle	La région que Wisej.NET a prévue pour cette opération de dessin; la mise en page peut consulter la taille cliente du contrôle, mais le dessin reste borné par le rectangle qui vous a été remis.
Helper GaugeGeometry	Une fonction pure d'un Rectangle et d'une valeur normalisée vers les limites de l'anneau, les angles de balayage, les points de l'aiguille et un rectangle de libellé, afin que les calculs puissent être vérifiés sans contrôle vivant.
Invalidate()	Le seul appel qui demande à Wisej.NET de refaire le rendu de la surface du contrôle; un setter de propriété l'émet exactement une fois, et uniquement lorsque le changement peut modifier un pixel.
Propriété des ressources	Le gestionnaire libère chaque Pen, Brush, Font, GraphicsPath et Bitmap qu'il crée, et ne libère jamais le Graphics fourni par le framework.
Budget de coût de dessin	Trois nombres par contrôle dessiné: le CPU serveur consacré au dessin, la mémoire serveur de l'image temporaire, et les octets que le navigateur doit récupérer et décoder.
AccessibleDescription	Porte la même mesure sous forme de texte, pour qu'une valeur qui n'existe que comme un arc dessiné reste disponible à un utilisateur qui ne verra jamais cet arc.

Construisez-le — choisissez votre assistant

Les deux boutons ouvrent un nouvel onglet avec le **même prompt prêt à l'emploi** déjà rempli. Le prompt oriente le modèle vers la spécification de ce module — un unique fichier Markdown qui contient la leçon, l'énoncé du lab et les livrables — pour qu'il travaille sur le vrai contenu au lieu de deviner de quoi parle le Module 2. Vous pouvez ouvrir vous-même [module-02.md](#) pour voir exactement ce qu'il lira.



Utiliser ChatGPT

Ouvre chatgpt.com avec le prompt de ce module déjà chargé. Fonctionne avec GPT-5 ou Codex ; toute offre capable de naviguer sur le web récupérera la spécification.



Utiliser Claude

Ouvre claude.ai avec le même prompt déjà chargé. Si vous utilisez Claude Code, vous pouvez plutôt le coller dans un terminal ouvert sur un dossier vide.



Envoyez-le tel quel et vous obtenez un projet, pas un extrait

Le prompt se termine en demandant toute la solution sous forme d'un unique [.zip](#) , avec les étapes pour la compiler et l'exécuter. Téléchargez-le, ouvrez la solution dans Visual Studio, appuyez sur F5 et VisualOperationsStudio s'ouvre sur [localhost](#) . Il vous faut le SDK .NET et le package NuGet [Wisej-4](#) ; tout le reste, le prompt demande à l'assistant de l'écrire lui-même.

Je suis le Module 2 - « Maîtriser l'événement Paint des contrôles » - du cours LearnWisej « Dessin et peinture dans Wisej.NET », et je veux construire avec toi l'exemple de ce module. Réponds-moi en français.

Commence par lire la spécification du module. C'est un unique fichier Markdown (en anglais) qui contient la leçon complète, l'énoncé du lab et les livrables :

<https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-02.md>

Construis ensuite un projet Wisej.NET 4 complet que je puisse ouvrir dans Visual Studio :

1. Projet « VisualOperationsStudio », C#, net8.0 ou ultérieur, package NuGet Wisej-4 4.1.x, hôte Kestrel qui appelle app.UseWisej().
2. Construis ce que demande le lab de la spécification : Ajoutez à VisualOperationsStudio un contrôle TelemetryGauge réutilisable qui dessine un anneau, des zones de seuil colorées, une aiguille, une valeur numérique et une légende. Dotez-le de propriétés Value, Minimum, Maximum, WarningThreshold et CriticalThreshold dont les setters bornent la valeur reçue, sortent immédiatement si rien n'a changé et appellent Invalidate() une seule fois. Placez les calculs de coordonnées dans un helper pur GaugeGeometry qui prend un Rectangle et renvoie les limites de l'anneau, les angles de balayage, les points de l'aiguille et le rectangle du libellé, puis dessinez à partir de ces valeurs avec des pens, des brushes et un GraphicsPath portés par des using, sans toucher au Graphics fourni. Gérez sans lever d'exception un contrôle de taille nulle, une plage inversée et une valeur hors plage, et exposez la mesure via AccessibleDescription pour qu'elle n'existe pas uniquement sous forme de pixels.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Contrôle TelemetryGauge avec des propriétés Value, Minimum, Maximum et de seuil qui bornent la valeur et appellent Invalidate() une fois par changement réel
 - Helper pur GaugeGeometry renvoyant les limites de l'anneau, les angles de balayage, les points de l'aiguille et le rectangle du libellé
 - Gestionnaire Paint qui libère chaque pen, brush, font et path qu'il crée et ne libère jamais e.Graphics
 - Garde-fous pour une taille nulle, une plage inversée et des valeurs hors plage, avec une AccessibleDescription portant la mesure
 - PaintCost.md indiquant la taille du redessin, sa fréquence et les cas où un contrôle théme coûterait moins cher
4. Suis la spécification partout où elle nomme une classe, une propriété ou un événement Wisej.NET - utilise exactement celui-là. Là où elle laisse le choix, dis-moi quelle API tu as choisie et pourquoi.
5. Garde l'autorité côté serveur : l'état et la validation vivent dans .NET, le navigateur affiche et remonte les informations. Ne déplace jamais une décision dans JavaScript parce que c'est plus simple là-bas.
6. Rends le résultat démontrable : je dois pouvoir l'exécuter et voir le comportement dont traite ce module, y compris ce qui se passe quand quelque chose tourne mal.

Avant d'écrire la moindre ligne de code, reformule avec tes propres mots ce que tu vas construire et où vivra chaque élément, pour que je puisse corriger la conception tant que cela ne coûte pas cher.

Quand le code est terminé, rassemble toute la solution dans un unique fichier .zip téléchargeable et donne-moi les étapes exactes pour la compiler et l'exécuter.

Enfin, réponds à ces questions de la spécification en t'appuyant sur ta propre

implémentation :

- Un collègue déplace les calculs de l'anneau et de l'aiguille directement dans le gestionnaire Paint parce que « ce n'est que quelques lignes ». Quelles deux choses deviennent ensuite impossibles à faire à bas coût, et laquelle des deux le Module 6 exige-t-il?
- Votre jauge est liée à une fiche machine qui porte aussi un nom d'opérateur et une date de maintenance. Si ces champs changent, le contrôle doit-il s'invalider? Répondez en termes de dépendances visuelles, et non en termes de modèle.
- La même jauge va figurer sur un tableau de bord qui en affiche soixante et se rafraîchit toutes les deux secondes. Que mesureriez-vous en premier, et que changeriez-vous si les mesures étaient mauvaises?

Pose-moi des questions sur tout ce qui est ambigu avant de commencer.

Puis relisez ce qui vous revient

Un code qui compile n'est pas encore un code juste. Les questions intéressantes sont des questions de conception : est-ce bien le contrôle Wisej.NET que la tâche exige, suit-il le cycle de vie que le framework vous offre, l'état de référence est-il sur le serveur ou a-t-il été laissé dans le navigateur parce que c'était pratique ? Lisez le résultat en relecteur, pas en client.

Ce module vous donne la checklist. Répondez à chaque point en vous appuyant sur le code que vous venez de recevoir :

- ✓ Un collègue déplace les calculs de l'anneau et de l'aiguille directement dans le gestionnaire Paint parce que « ce n'est que quelques lignes ». Quelles deux choses deviennent ensuite impossibles à faire à bas coût, et laquelle des deux le Module 6 exige-t-il?
- ✓ Votre jauge est liée à une fiche machine qui porte aussi un nom d'opérateur et une date de maintenance. Si ces champs changent, le contrôle doit-il s'invalider? Répondez en termes de dépendances visuelles, et non en termes de modèle.
- ✓ La même jauge va figurer sur un tableau de bord qui en affiche soixante et se rafraîchit toutes les deux secondes. Que mesureriez-vous en premier, et que changeriez-vous si les mesures étaient mauvaises?

Tout ce que vous avez dû corriger mérite d'être noté. Une courte note des conventions et des décisions retenues, collée dans le premier message de votre prochaine session, vaut plus que n'importe quel prompt astucieux.



Pas encore d'exemple de référence pour ce cours — le vôtre est le premier

Le [dépôt des exemples LearnWisej](#) contient des solutions complètes pour la plupart des cours, mais pas encore pour celui-ci. Il n'y a donc rien à quoi vous comparer et aucun raccourci : la spécification et votre propre relecture sont votre seul filet de sécurité. Gardez ce que vous construisez — et si le résultat est réussi, nous aimerions le voir.

Trois prompts pour aller plus loin

Orientez votre assistant vers le projet que vous venez de construire et essayez ceux-ci. Chacun est un exercice de relecture et d'extension, pas une réécriture.

- ✓ **Prouvez que ça fonctionne vraiment.** *"Passe en revue ce projet livrable par livrable par rapport à la spécification du module sur <https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-02.md> et montre-moi la preuve que chacun est réellement terminé — le chemin dans le code, où cliquer et ce que je dois voir. Là où un livrable n'est que partiellement satisfait, dis-le clairement au lieu de le défendre. Ensuite, ajoute ce qui manque."*
- ✓ **Relisez-le comme une pull request.** *"Fais-moi parcourir ce code VisualOperationsStudio comme si tu relisais la pull request d'un collègue et que tu ne l'avais pas écrit. Où vit l'état, que se passe-t-il de bout en bout lors d'une action de l'utilisateur, que renverrais-tu pour correction, et quelles parties ne serais-tu pas à l'aise de montrer à un client ? Sois précis et cite les lignes dont tu parles — ne me résume pas la conception."*
- ✓ **Remplacez une pièce sans déplacer les jointures.** *"Choisis la partie de ce projet VisualOperationsStudio qui remplace le plus manifestement quelque chose de réel — une source de données factice, un stockage en mémoire, un service simplifié — et remplace-la par une implémentation digne de la production. Le contrat de part et d'autre ne doit pas changer : mêmes types, mêmes appels, même comportement en cas d'échec. Dis-moi ce que l'implémentation réelle t'a obligé à gérer et que la version provisoire ne gérait pas."*



Emportez-le dans le lab

Le lab pratique qui clôt ce module demande les mêmes livrables que ceux que vous venez de faire produire à l'assistant. Servez-vous de ce que vous avez ici comme forme de votre réponse et comme preuve à l'appui — pas comme la réponse elle-même. Le lab est noté sur votre raisonnement, et cette partie doit venir de vous.