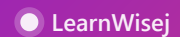


Exercice de code avec l'IA

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.



Écrire le code, c'est désormais la moitié facile.

Obtenir d'un assistant moderne du code qui fonctionne pour « Wisej.Web.Canvas et le modèle Canvas 2D de HTML5 » ne demande qu'un bon prompt. Le juger, le corriger et l'étendre demande tout ce que ce module a couvert. C'est sur ce partage que repose l'exercice : vous apportez la spécification et la relecture, l'assistant apporte la frappe.

Ce que vous demandez

Construisez dans VisualOperationsStudio une page bac à sable Canvas qui porte une série d'exemples Canvas 2D du navigateur vers l'API Wisej.Web.Canvas documentée. Pilotez le tout depuis une méthode DrawPlayground appelée par l'événement Redraw du Canvas, afin que la scène revienne après un redimensionnement, et commencez-la par un ClearRect sur toute la surface. Dessinez un tracé en contour, des rectangles pleins, une légende placée, un dégradé linéaire et un dégradé radial, puis encadrez chaque section transformée par Save() et Restore() et démontrez Translate, Scale, Rotate et SetTransform. Ajoutez une région découpée, une section globalAlpha, une ligne pointillée et une ombre, portez un exemple du navigateur qui pivote de $\text{Math.PI} / 4$ vers la valeur en degrés attendue par Wisej, et laissez LiveUpdate désactivé pour le rendu principal tout en le montrant dans un petit exemple progressif.

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

- ✓ Bac à sable Canvas dont toute la scène est dessinée par une méthode DrawPlayground appelée depuis Redraw
- ✓ Tracés, rectangles, légende placée, dégradé linéaire et dégradé radial n'utilisant que des membres documentés de Wisej.Web.Canvas
- ✓ Save() et Restore() autour de chaque section Translate, Scale, Rotate et SetTransform
- ✓ Découpage, alpha, ligne pointillée et ombre, plus un portage radians vers degrés écrit noir sur blanc dans un commentaire
- ✓ Une démonstration de LiveUpdate tenue à l'écart du rendu principal, avec une note sur les cas où elle est appropriée

Pourquoi il est conçu ainsi

Tout ce que VisualOperationsStudio a dessiné jusqu'ici a été produit sur le serveur: la jauge du Module 2 et les cellules Health et Trend du Module 3 sont rendues avec System.Drawing et parviennent à l'utilisateur

sous forme d'images. La vue de topologie dont l'application a désormais besoin, ce sont des dizaines de petits nœuds, d'arêtes et de libellés qu'un utilisateur finira par déplacer, zoomer et sélectionner — exactement le travail qu'un canvas de navigateur fait bien.

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.Canvas	Un contrôle Wisej qui expose à C# la surface 2D en mode immédiat du navigateur; ses appels sont des commandes de dessin que le navigateur exécute, et non des opérations System.Drawing.Graphics.
événement Redraw	Le point d'accroche que la surface déclenche lorsqu'elle a de nouveau besoin de son contenu; comme Wisej.NET ne conserve pas les commandes, le gestionnaire doit reconstruire toute la scène.
correspondance PascalCase	Les membres se transposent par la casse — fillRect() vers FillRect(), strokeStyle vers StrokeStyle — tandis que les valeurs CSS deviennent des types.NET comme Color, Rectangle, Font et Image.
angles en degrés	Wisej.NET documente les arguments d'angle en degrés par simplicité, là où le navigateur utilise des radians, ce qui change chaque appel Arc() et Rotate() porté.
Save() / Restore()	Empilent et dépilent tout l'état de rendu — styles, transformation, découpage, alpha, réglages de ligne et de texte — pour qu'un changement local ne puisse pas fuiter vers la forme suivante.
TextFont, TextAlign, TextBaseline	Le pendant Wisej.NET des font, textAlign et textBaseline du navigateur; la police est une véritable System.Drawing.Font.
LiveUpdate	À false par défaut: les commandes voyagent alors avec la mise à jour d'interface normale de la requête; à true, chaque appel est poussé immédiatement, ce qui convient à un dévoilement progressif et ruine une boucle d'animation.

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 4. Vous pouvez ouvrir vous-même [module-04.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



Utiliser Claude

Ouvre [claude.ai](#) avec le même prompt déjà chargé. Si vous utilisez Claude Code,

naviguer sur le web récupérera la spécification.

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 4 - « Wisej.Web.Canvas et le modèle Canvas 2D de HTML5 » - 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-04.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 : Construisez dans VisualOperationsStudio une page bac à sable Canvas qui porte une série d'exemples Canvas 2D du navigateur vers l'API Wisej.Web.Canvas documentée. Pilotez le tout depuis une méthode DrawPlayground appelée par l'événement Redraw du Canvas, afin que la scène revienne après un redimensionnement, et commencez-la par un ClearRect sur toute la surface. Dessinez un tracé en contour, des rectangles pleins, une légende placée, un dégradé linéaire et un dégradé radial, puis encadrez chaque section transformée par Save() et Restore() et démontrez Translate, Scale, Rotate et SetTransform. Ajoutez une région découpée, une section globalAlpha, une ligne pointillée et une ombre, portez un exemple du navigateur qui pivote de $\text{Math.PI} / 4$ vers la valeur en degrés attendue par Wisej, et laissez LiveUpdate désactivé pour le rendu principal tout en le montrant dans un petit exemple progressif.
3. Produis ces livrables et indique-moi où se trouve chacun d'eux dans la solution :
 - Bac à sable Canvas dont toute la scène est dessinée par une méthode DrawPlayground appelée depuis Redraw
 - Tracés, rectangles, légende placée, dégradé linéaire et dégradé radial n'utilisant que des membres documentés de Wisej.Web.Canvas
 - Save() et Restore() autour de chaque section Translate, Scale, Rotate et SetTransform
 - Découpage, alpha, ligne pointillée et ombre, plus un portage radians vers degrés écrit noir sur blanc dans un commentaire
 - Une démonstration de LiveUpdate tenue à l'écart du rendu principal, avec une note sur les cas où elle est appropriée
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 :

- Votre bac à sable paraît correct jusqu'à ce qu'un collègue redimensionne le navigateur,

après quoi seule la dernière forme dessinée reste visible. Que cela vous apprend-il sur l'origine du reste de la scène?

- Vous avez besoin de la largeur en pixels d'une légende pour dimensionner une pastille à sa mesure. L'API Canvas de Wisej.NET ne liste aucun membre de mesure de texte — décrivez deux façons d'obtenir malgré tout la mise en page voulue, et ce que chacune coûte.
- Un relecteur demande pourquoi le découpage de votre section d'état est encadré par Save() et Restore() alors que les formes qui suivent sont dessinées tout à fait ailleurs. Répondez-lui.

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 :

- ✓ Votre bac à sable paraît correct jusqu'à ce qu'un collègue redimensionne le navigateur, après quoi seule la dernière forme dessinée reste visible. Que cela vous apprend-il sur l'origine du reste de la scène?
- ✓ Vous avez besoin de la largeur en pixels d'une légende pour dimensionner une pastille à sa mesure. L'API Canvas de Wisej.NET ne liste aucun membre de mesure de texte — décrivez deux façons d'obtenir malgré tout la mise en page voulue, et ce que chacune coûte.
- ✓ Un relecteur demande pourquoi le découpage de votre section d'état est encadré par Save() et Restore() alors que les formes qui suivent sont dessinées tout à fait ailleurs. Répondez-lui.

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.

- ✓ **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."

- ✓ **Faites-le résister aux mauvais jours.** *"Examine ce projet VisualOperationsStudio pour repérer les modes de défaillance qu'il ne gère pas encore — une saisie erronée ou manquante, une dépendance lente ou indisponible, un traitement qui aboutit après que l'utilisateur a changé de page, et deux actions qui se chevauchent. Pour chacun, dis-moi d'abord s'il s'agit d'un risque réel dans Wisej.NET 4 et pourquoi, et seulement ensuite écris la correction. Là où tu estimes qu'aucune gestion n'est nécessaire, dis-le et explique ton raisonnement au lieu d'ajouter du code défensif."*
- ✓ **Ajoutez le deuxième cas.** *"Étends ce projet VisualOperationsStudio avec un deuxième cas qui suit le même modèle que le premier — une autre entité, un autre écran ou un autre contrôle de même forme, câblé de la même manière. Extrais dans une classe de base ou un service ce qui est réellement partagé, mais seulement là où cela se justifie ; deux choses similaires n'appellent pas toujours une abstraction. Montre-moi le diff et justifie chaque élément que tu as extrait."*



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.