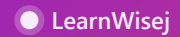


KI-Coding-Übung

Erstellen Sie das Beispielprojekt VisualOperationsStudio dieses Moduls mit ChatGPT oder Claude aus einem fertigen Prompt, der das Modell auf die Spezifikation des Moduls verweist – und prüfen, starten und erweitern Sie dann, was Sie erhalten.



Den Code zu schreiben ist heute die leichtere Hälfte.

Funktionierenden Code für „Wisej.Web.Canvas und das HTML5-Canvas-2D-Modell“ bekommen Sie von einem modernen Assistenten mit einem guten Prompt. Ihn zu beurteilen, zu korrigieren und zu erweitern, erfordert alles, was dieses Modul behandelt hat. Genau auf dieser Aufteilung baut die Übung auf: Sie liefern Spezifikation und Review, der Assistent die Tipparbeit.

Worum Sie bitten

Bauen Sie in VisualOperationsStudio eine Canvas-Spielwiese, die eine Reihe von Canvas-2D-Beispielen aus dem Browser auf die dokumentierte Wisej.Web.Canvas-API portiert. Steuern Sie alles über eine Methode DrawPlayground, die vom Redraw-Event der Canvas aufgerufen wird, damit die Szene nach einer Größenänderung zurückkehrt, und beginnen Sie sie mit ClearRect über die gesamte Fläche. Rendern Sie einen umrissenen Pfad, gefüllte Rechtecke, eine platzierte Beschriftung, einen linearen und einen radialen Farbverlauf, klammern Sie dann jeden transformierten Abschnitt in Save() und Restore() und führen Sie Translate, Scale, Rotate und SetTransform vor. Ergänzen Sie einen beschnittenen Bereich, einen Alpha-Abschnitt, eine gestrichelte Linie und einen Schatten, portieren Sie ein Browser-Beispiel, das um $\text{Math.PI} / 4$ dreht, in den Gradwert, den Wisej erwartet, und lassen Sie LiveUpdate für das Hauptrendering ausgeschaltet, während Sie es in einem kleinen fortlaufenden Beispiel zeigen.

Das Modul verlangt diese Ergebnisse, und der Prompt unten fordert genau diese vom Assistenten an:

- ✓ Canvas-Spielwiese, deren gesamte Szene von einer aus Redraw aufgerufenen Methode DrawPlayground gezeichnet wird
- ✓ Pfade, Rechtecke, eine platzierte Beschriftung, ein linearer und ein radialer Farbverlauf ausschließlich mit dokumentierten Wisej.Web.Canvas-Mitgliedern
- ✓ Save() und Restore() um jeden Abschnitt mit Translate, Scale, Rotate und SetTransform
- ✓ Clipping, Alpha, eine gestrichelte Linie und ein Schatten, dazu eine als Kommentar ausgeschriebene Umrechnung von Radiant in Grad
- ✓ Eine LiveUpdate-Demonstration getrennt vom Hauptrendering, mit einer Notiz, wann sie angebracht ist

Warum es so aufgebaut ist

Alles, was VisualOperationsStudio bisher gezeichnet hat, entstand auf dem Server: Das Gauge aus Modul 2 und die Zellen Health und Trend aus Modul 3 werden mit `System.Drawing` gerendert und erreichen die Benutzerin als Bilder. Die Topologieansicht, die die Anwendung als Nächstes braucht, besteht aus Dutzenden kleiner Knoten, Kanten und Beschriftungen, die eine Benutzerin irgendwann verschieben, zoomen und auswählen wird – genau die Arbeit, die eine Browser-Canvas gut erledigt.

Diese Begriffe muss der Assistent richtig verwenden. Geht die Antwort nachlässig mit ihnen um, hat er das Modul nicht verstanden:

Begriff	Was er hier bedeutet
Wisej.Web.Canvas	Ein Wisej-Control, das die 2D-Oberfläche des Browsers im Immediate Mode für C# zugänglich macht; seine Aufrufe sind Zeichenbefehle, die der Browser ausführt, keine Operationen auf <code>System.Drawing.Graphics</code> .
Redraw -Event	Der Haken, den die Oberfläche auslöst, wenn sie ihren Inhalt erneut braucht; weil Wisej.NET die Befehle nicht aufbewahrt, muss der Handler die gesamte Szene neu aufbauen.
Entsprechung in PascalCase	Die Member bilden sich über die Schreibweise ab — <code>fillRect()</code> auf <code>FillRect()</code> , <code>strokeStyle</code> auf <code>StrokeStyle</code> —, während aus CSS-Werten.NET-Typen wie <code>Color</code> , <code>Rectangle</code> , <code>Font</code> und <code>Image</code> werden.
Winkel in Grad	Wisej.NET dokumentiert Winkelargumente der Einfachheit halber in Grad, wo der Browser Radiant verwendet – das ändert jeden portierten Aufruf von <code>Arc()</code> und <code>Rotate()</code> .
Save() / Restore()	Legen den gesamten Rendering-Zustand ab und holen ihn zurück — Stile, Transformation, Clipping, Alpha, Linien- und Texteeinstellungen —, damit eine lokale Änderung nicht in die nächste Form überläuft.
TextFont, TextAlign, TextBaseline	Der Wisej.NET-Ersatz für <code>font</code> , <code>textAlign</code> und <code>textBaseline</code> des Browsers; die Schrift ist ein echter <code>System.Drawing.Font</code> .
LiveUpdate	Standardmäßig <code>false</code> , sodass die Befehle mit der normalen UI-Aktualisierung der Anfrage reisen; <code>true</code> sendet jeden Aufruf sofort, was zu einer schrittweisen Enthüllung passt und eine Animationsschleife ruiniert.

Selbst bauen – wählen Sie Ihren Assistenten

Beide Buttons öffnen einen neuen Tab, in dem **derselbe fertige Prompt** bereits eingetragen ist. Der Prompt verweist das Modell auf die Spezifikation dieses Moduls – eine einzelne Markdown-Datei mit der Lektion, der Lab-Aufgabe und den geforderten Ergebnissen –, sodass es mit dem echten Material arbeitet, statt zu raten, worum es in Modul 4 geht. Sie können [module-04.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden



Claude verwenden

Öffnet chatgpt.com mit dem Prompt dieses Moduls. Funktioniert mit GPT-5 oder Codex; jeder Tarif mit Websuche ruft die Spezifikation ab.

Öffnet claude.ai mit demselben Prompt. Wenn Sie Claude Code nutzen, können Sie ihn stattdessen in einem leeren Ordner in ein Terminal einfügen.



Unverändert abgeschickt, erhalten Sie ein Projekt, kein Code-Fragment

Der Prompt verlangt am Ende die gesamte Projektmappe als einzelne [.zip](#) -Datei mit Anleitung zum Kompilieren und Starten. Laden Sie sie herunter, öffnen Sie die Projektmappe in Visual Studio, drücken Sie F5, und VisualOperationsStudio startet auf [localhost](#) . Sie benötigen das .NET SDK und das NuGet-Paket [Wisej-4](#) ; alles andere soll der Assistent laut Prompt selbst schreiben.

Ich arbeite gerade Modul 4 – „Wisej.Web.Canvas und das HTML5-Canvas-2D-Modell“ – des LearnWisej-Kurses „Zeichnen und Grafik in Wisej.NET“ durch und möchte das Beispielprojekt des Moduls gemeinsam mit dir erstellen. Bitte antworte auf Deutsch.

Lies zuerst die Modulspezifikation. Sie ist eine einzelne Markdown-Datei (auf Englisch) mit der vollständigen Lektion, der Lab-Aufgabe und den geforderten Ergebnissen:

<https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-04.md>

Erstelle dann ein vollständiges Wisej.NET-4-Projekt, das ich in Visual Studio öffnen kann:

1. Projekt „VisualOperationsStudio“, C#, net8.0 oder neuer, NuGet-Paket Wisej-4 4.1.x, Kestrel-Host, der `app.UseWisej()` aufruft.
2. Setze um, was das Lab der Spezifikation verlangt: Bauen Sie in `VisualOperationsStudio` eine Canvas-Spielwiese, die eine Reihe von Canvas-2D-Beispielen aus dem Browser auf die dokumentierte `Wisej.Web.Canvas-API` portiert. Steuern Sie alles über eine Methode `DrawPlayground`, die vom `Redraw`-Event der Canvas aufgerufen wird, damit die Szene nach einer Größenänderung zurückkehrt, und beginnen Sie sie mit `ClearRect` über die gesamte Fläche. Rendern Sie einen umrissenen Pfad, gefüllte Rechtecke, eine platzierte Beschriftung, einen linearen und einen radialen Farbverlauf, klammern Sie dann jeden transformierten Abschnitt in `Save()` und `Restore()` und führen Sie `Translate`, `Scale`, `Rotate` und `SetTransform` vor. Ergänzen Sie einen beschnittenen Bereich, einen Alpha-Abschnitt, eine gestrichelte Linie und einen Schatten, portieren Sie ein Browser-Beispiel, das um $\text{Math.PI} / 4$ dreht, in den Gradwert, den Wisej erwartet, und lassen Sie `LiveUpdate` für das Hauptrendering ausgeschaltet, während Sie es in einem kleinen fortlaufenden Beispiel zeigen.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - Canvas-Spielwiese, deren gesamte Szene von einer aus `Redraw` aufgerufenen Methode `DrawPlayground` gezeichnet wird
 - Pfade, Rechtecke, eine platzierte Beschriftung, ein linearer und ein radialer Farbverlauf ausschließlich mit dokumentierten `Wisej.Web.Canvas`-Membern
 - `Save()` und `Restore()` um jeden Abschnitt mit `Translate`, `Scale`, `Rotate` und `SetTransform`
 - Clipping, Alpha, eine gestrichelte Linie und ein Schatten, dazu eine als Kommentar ausgedruckte Umrechnung von Radiant in Grad
 - Eine `LiveUpdate`-Demonstration getrennt vom Hauptrendering, mit einer Notiz, wann sie angebracht ist
4. Halte dich an die Spezifikation, wo immer sie eine `Wisej.NET`-Klasse, -Eigenschaft oder ein -Event nennt – verwende genau diese. Wo sie eine Wahl offenlässt, sag mir, welche API du gewählt hast und warum.
5. Der Server bleibt maßgeblich: Zustand und Validierung leben in .NET, der Browser stellt dar und meldet zurück. Verlagere nie eine Entscheidung nach JavaScript, nur weil es dort einfacher ist.
6. Mach das Ergebnis vorführbar: Ich muss es starten und das Verhalten sehen können, um das es in diesem Modul geht – auch, was passiert, wenn etwas schiefgeht.

Bevor du Code schreibst, fasse in eigenen Worten zusammen, was du bauen wirst und wo jedes Teil liegen wird, damit ich den Entwurf korrigieren kann, solange das noch wenig kostet.

Wenn der Code fertig ist, packe die gesamte Projektmappe in eine einzelne .zip-Datei zum Herunterladen und nenne mir die genauen Schritte, um sie zu kompilieren und auszuführen.

Beantworte zum Schluss diese Fragen aus der Spezifikation und nutze dabei deine eigene Implementierung als Beleg:

- Ihre Spielwiese sieht richtig aus, bis eine Kollegin die Browsergröße ändert – danach ist

nur noch die zuletzt gezeichnete Form sichtbar. Was sagt Ihnen das darüber, woher der Rest der Szene kam?

- Sie brauchen die Pixelbreite einer Beschriftung, damit ein Badge passend bemessen werden kann. Die Canvas-API von Wisej.NET führt keinen Member zur Textvermessung auf – beschreiben Sie zwei Wege, das gewünschte Layout dennoch zu erreichen, und was jeder davon kostet.
- Ein Reviewer fragt, warum das Clipping in Ihrem Zustandsabschnitt in `Save()` und `Restore()` eingeklammert ist, obwohl die Formen danach an ganz anderer Stelle gezeichnet werden. Antworten Sie ihm.

Frag mich nach allem, was unklar ist, bevor du anfängst.

Prüfen Sie dann, was zurückkommt

Code, der kompiliert, ist noch lange kein richtiger Code. Die spannenden Fragen sind Designfragen: Ist das wirklich das Wisej.NET-Control, das die Aufgabe verlangt? Folgt der Code dem Lebenszyklus, den das Framework vorgibt? Liegt der maßgebliche Zustand auf dem Server – oder wurde er im Browser abgelegt, weil das bequemer war? Lesen Sie das Ergebnis als Reviewer, nicht als Kunde.

Die Checkliste liefert Ihnen dieses Modul. Beantworten Sie jeden Punkt anhand des Codes, den Sie gerade erhalten haben:

- ✓ Ihre Spielwiese sieht richtig aus, bis eine Kollegin die Browsergröße ändert – danach ist nur noch die zuletzt gezeichnete Form sichtbar. Was sagt Ihnen das darüber, woher der Rest der Szene kam?
- ✓ Sie brauchen die Pixelbreite einer Beschriftung, damit ein Badge passend bemessen werden kann. Die Canvas-API von Wisej.NET führt keinen Member zur Textvermessung auf — beschreiben Sie zwei Wege, das gewünschte Layout dennoch zu erreichen, und was jeder davon kostet.
- ✓ Ein Reviewer fragt, warum das Clipping in Ihrem Zustandsabschnitt in `Save()` und `Restore()` eingeklammert ist, obwohl die Formen danach an ganz anderer Stelle gezeichnet werden. Antworten Sie ihm.

Alles, was Sie korrigieren mussten, sollten Sie festhalten. Eine kurze Notiz zu den Konventionen und Entscheidungen, auf die Sie sich festgelegt haben – eingefügt in die erste Nachricht Ihrer nächsten Sitzung –, ist mehr wert als jeder noch so clevere Prompt.



Für diesen Kurs gibt es noch kein Referenzbeispiel – Ihres ist das erste

Das [LearnWisej-Beispiel-Repository](#) enthält ausgearbeitete Builds für die meisten Kurse, für diesen aber noch nicht. Es gibt also nichts zum Vergleichen und keine Abkürzung: Die Spezifikation und Ihr eigenes Review sind das ganze Sicherheitsnetz. Behalten Sie, was Sie bauen – und wenn es gut wird, würden wir es gern sehen.

Drei Prompts für den nächsten Schritt

Richten Sie Ihren Assistenten auf das Projekt, das Sie gerade gebaut haben, und probieren Sie diese Prompts aus. Jeder ist eine Übung zum Prüfen und Erweitern, kein Neuschreiben.

- ✓ **Tauschen Sie ein Teil aus, ohne die Schnittstellen zu verschieben.** *"Wähle den Teil dieses VisualOperationsStudio-Projekts, der am offensichtlichsten nur ein Platzhalter für etwas Echtes ist – eine Fake-Datenquelle, ein In-Memory-Speicher, ein vereinfachter Service – und ersetze ihn durch eine praxisnahe Implementierung. Der Vertrag auf beiden Seiten darf sich nicht ändern: dieselben Typen, dieselben Aufrufe, dasselbe Verhalten im Fehlerfall. Sag mir, was du durch die echte Implementierung behandeln musstest, was der Platzhalter nicht behandelt hat."*
- ✓ **Machen Sie es fit für einen schlechten Tag.** *"Untersuche dieses VisualOperationsStudio-Projekt auf Fehlerszenarien, die es noch nicht behandelt – fehlerhafte oder fehlende Eingaben, eine Abhängigkeit, die langsam oder nicht erreichbar ist, Arbeit, die eintrifft, nachdem der Benutzer die Seite verlassen hat, und zwei Aktionen, die sich gegenseitig in die Quere kommen. Sag mir für jedes zuerst, ob es in Wisej.NET 4 ein echtes Risiko ist und warum, und schreib erst dann die Korrektur. Wo du entscheidest, dass nichts behandelt werden muss, sag das und begründe es, statt defensiven Code hinzuzufügen."*
- ✓ **Ergänzen Sie den zweiten Fall.** *"Erweitere dieses VisualOperationsStudio-Projekt um einen zweiten Fall, der demselben Muster folgt wie der erste – eine weitere Entität, Maske oder ein Control derselben Form, auf dieselbe Weise verdrahtet. Lagere wirklich Gemeinsames in eine Basisklasse oder einen Service aus, aber nur dort, wo es sich lohnt; zwei ähnliche Dinge brauchen nicht immer eine Abstraktion. Zeig mir den Diff und begründe jede Auslagerung."*



Nehmen Sie es mit ins Lab

Das praktische Lab am Ende dieses Moduls verlangt dieselben Ergebnisse, die Sie gerade vom Assistenten haben erstellen lassen. Nutzen Sie, was Sie hier haben, als Gerüst für Ihre Antwort und als Beleg dafür – nicht als die Antwort selbst. Bewertet wird im Lab Ihre Argumentation, und dieser Teil muss von Ihnen stammen.