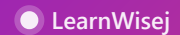


KI-Coding-Übung

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.



Dasselbe Modul – so gebaut, wie Sie in der Praxis wirklich bauen.

Alle anderen Abschnitte dieses Moduls erklären, wie „Interaktives Canvas: Zustand, Eingaben, Trefferprüfung und Performance“ funktioniert. Dieser fordert Sie auf, es herzustellen: indem Sie einen Assistenten mit der Spezifikation des Moduls steuern und dann prüfen, ob das Ergebnis läuft und tut, was die Lab-Aufgabe beschreibt. Es ist der Arbeitsablauf, den Sie in echten Projekten nutzen werden – geübt an etwas, dessen richtige Lösung Sie bereits kennen.

Worum Sie bitten

Machen Sie aus der Canvas-Spielwiese einen Topologie-Editor für VisualOperationsStudio. Erstellen Sie die Klassen NodeModel und EdgeModel mit stabilen IDs und Geometrie in Weltkoordinaten, halten Sie die Szene in einem TopologyScene-Objekt pro Sitzung statt in einem statischen Feld und lassen Sie RenderScene die einzige zeichnende Methode bleiben. Ergänzen Sie ToWorld- und ToScreen-Umrechnungen für den aktuellen Verschiebe-Offset und den Zoomfaktor, behandeln Sie dann MouseDown, um die Knotengrenzen in Weltkoordinaten per Trefferprüfung zu ermitteln und Selected zu setzen, MouseMove, um den ausgewählten Knoten zu ziehen oder den Viewport zu verschieben, und MouseUp, um die Geste zu beenden – jeweils so, dass sie den Zustand ändern und anschließend RenderScene aufrufen. Ergänzen Sie Zoomen per Mausrad oder Button um die Zeigerposition herum, überspringen Sie beim Rendern Knoten außerhalb des sichtbaren Rechtecks, geben Sie Tastaturnutzern einen Weg, einen Knoten auszuwählen und zu verschieben, und halten Sie fest, wie viele Anfragen ein Ziehvorgang erzeugt und ab welchem Punkt Sie die Schleife nach JavaScript im Client verlagern würden.

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

- ✓ NodeModel, EdgeModel und eine TopologyScene pro Sitzung mit stabilen IDs und ohne veränderlichen statischen Zustand
- ✓ ToWorld- und ToScreen-Umrechnungen für die Trefferprüfung, damit die Auswahl nach Verschieben und Zoomen korrekt bleibt
- ✓ MouseDown-, MouseMove- und MouseUp-Handler, die den Zustand ändern und dann die eine Methode RenderScene aufrufen
- ✓ Viewport-Culling, das Knoten außerhalb des sichtbaren Rechtecks überspringt, dazu ein Tastaturweg zum Auswählen und Verschieben
- ✓ InteractionNotes.md mit den während eines Ziehvorgangs gemessenen Anfragen und der Grenze, ab der Sie auf den Client umstellen würden

Warum es so aufgebaut ist

Modul 4 hat VisualOperationsStudio mit einer Canvas-Spielwiese zurückgelassen: einer Methode DrawPlayground, die das Redraw -Event aufruft und jeden Pfad, jeden Farbverlauf und jede Transformation von Grund auf neu aufbaut. Sie ist korrekt – und völlig regungslos. Die Betriebsmannschaft möchte auf eine Pumpe klicken, sie an eine sinnvolle Stelle ziehen, über die Anlage schwenken und in eine Gruppe hineinzoomen — und eine Zeichenmethode kann keine dieser Anfragen beantworten, denn die Browser-Canvas hält Pixel, und Pixel wissen nicht, welche Form eine Pumpe ist. In dem Moment, in dem eine Oberfläche interaktiv wird, hört sie auf, ein Renderer zu sein, und wird zu einer kleinen Zustandsmaschine, die nebenbei rendert.

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
Modell, Eingabe, Zustand, Rendering	Vier Schichten mit einer Regel dazwischen: Eingaben ändern den Zustand, und erst danach macht eine einzige Render-Methode aus dem aktuellen Zustand Zeichenaufrufe.
Szenenobjekte	C#-Klassen wie NodeModel und EdgeModel, die stabile ID, Grenzen in Weltkoordinaten, Status und das Selected -Kennzeichen tragen – all das, was Pixel nur andeuten können.
Szenenzustand pro Sitzung	Die Szene gehört der Sitzung einer Benutzerin; eine veränderliche static -Liste würde jedem Besucher dieselben Knoten und dieselbe Auswahl geben.
Eingabe-Events von Control	Canvas leitet sich von Control ab, also stehen Mouse-Down/Move/Up, Click, Double Click sowie die Touch- und Gestenereignisse wie TouchStart, Tap, Pinch, Rotate, Swipe und Track zur Verfügung.
Modellbasierte Trefferprüfung	Die dokumentierte Canvas-API von Wisej bietet weder isPointInPath noch isPointInStroke, also wird „worauf hat die Benutzerin geklickt?“ beantwortet, indem ein Punkt mit Geometrie verglichen wird, die Ihnen gehört.
Umrechnung Bildschirm nach Welt	Wendet das Rendering Bildschirm = Welt $\times$ Zoom + Verschiebung an, dann muss jede eingehende Zeigerkoordinate durch die Umkehrung laufen, bevor sie mit irgendetwas im Modell verglichen wird.
Viewport-Culling	Szenenobjekte überspringen, deren Grenzen das sichtbare Weltrechteck nicht schneiden, sodass eine große Topologie nur das kostet, was tatsächlich auf dem Bildschirm ist.

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 5 geht. Sie können [module-05.md](#) selbst öffnen, um genau zu sehen, was es lesen wird.



ChatGPT verwenden

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



Claude verwenden

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 5 – „Interaktives Canvas: Zustand, Eingaben, Trefferprüfung und Performance“ – 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-05.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: Machen Sie aus der Canvas-Spielwiese einen Topologie-Editor für VisualOperationsStudio. Erstellen Sie die Klassen `NodeModel` und `EdgeModel` mit stabilen IDs und Geometrie in Weltkoordinaten, halten Sie die Szene in einem `TopologyScene`-Objekt pro Sitzung statt in einem statischen Feld und lassen Sie `RenderScene` die einzige zeichnende Methode bleiben. Ergänzen Sie `ToWorld`- und `ToScreen`-Umrechnungen für den aktuellen Verschiebe-Offset und den Zoomfaktor, behandeln Sie dann `MouseDown`, um die Knotengrenzen in Weltkoordinaten per Trefferprüfung zu ermitteln und `Selected` zu setzen, `MouseMove`, um den ausgewählten Knoten zu ziehen oder den Viewport zu verschieben, und `MouseUp`, um die Geste zu beenden – jeweils so, dass sie den Zustand ändern und anschließend `RenderScene` aufrufen. Ergänzen Sie Zoomen per Mausrad oder Button um die Zeigerposition herum, überspringen Sie beim Rendern Knoten außerhalb des sichtbaren Rechtecks, geben Sie Tastaturnutzern einen Weg, einen Knoten auszuwählen und zu verschieben, und halten Sie fest, wie viele Anfragen ein Ziehvorgang erzeugt und ab welchem Punkt Sie die Schleife nach JavaScript im Client verlagern würden.
3. Liefere diese Ergebnisse und sag mir, wo sich jedes davon in der Projektmappe befindet:
 - `NodeModel`, `EdgeModel` und eine `TopologyScene` pro Sitzung mit stabilen IDs und ohne veränderlichen statischen Zustand
 - `ToWorld`- und `ToScreen`-Umrechnungen für die Trefferprüfung, damit die Auswahl nach Verschieben und Zoomen korrekt bleibt
 - `MouseDown`-, `MouseMove`- und `MouseUp`-Handler, die den Zustand ändern und dann die eine Methode `RenderScene` aufrufen
 - Viewport-Culling, das Knoten außerhalb des sichtbaren Rechtecks überspringt, dazu ein Tastaturweg zum Auswählen und Verschieben
 - `InteractionNotes.md` mit den während eines Ziehvorgangs gemessenen Anfragen und der Grenze, ab der Sie auf den Client umstellen würden
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:

- Eine Kollegin meldet, dass die Auswahl funktioniert, bis sie hineinzoomt – danach landen Klicks auf der falschen Pumpe oder auf nichts. Am Zeichencode hat sich nichts geändert. Welche einzelne Umrechnung fehlt, und warum bleibt der Defekt bei Zoom 1,0 ohne Verschiebung unsichtbar?
- Zwei Bedienende öffnen die Topologie gleichzeitig, und jede sieht den Knoten der anderen unter dem eigenen Cursor wandern. Worin wurde die Szene gespeichert, und was sagt Ihnen das darüber, wo interaktiver Zustand in einem serverseitigen Webframework hingehört?
- Ihr Ziehvorgang erzeugt für jede Zeigerbewegung eine Anfrage. Ab welchem gemessenen Punkt würden Sie aufhören, die Serverschleife zu optimieren, und die Interaktion stattdessen nach JavaScript im Client verlagern – und was müsste trotzdem auf dem Server maßgeblich bleiben?

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:

- ✓ Eine Kollegin meldet, dass die Auswahl funktioniert, bis sie hineinzoomt – danach landen Klicks auf der falschen Pumpe oder auf nichts. Am Zeichencode hat sich nichts geändert. Welche einzelne Umrechnung fehlt, und warum bleibt der Defekt bei Zoom 1,0 ohne Verschiebung unsichtbar?
- ✓ Zwei Bedienende öffnen die Topologie gleichzeitig, und jede sieht den Knoten der anderen unter dem eigenen Cursor wandern. Worin wurde die Szene gespeichert, und was sagt Ihnen das darüber, wo interaktiver Zustand in einem serverseitigen Webframework hingehört?
- ✓ Ihr Ziehvorgang erzeugt für jede Zeigerbewegung eine Anfrage. Ab welchem gemessenen Punkt würden Sie aufhören, die Serverschleife zu optimieren, und die Interaktion stattdessen nach JavaScript im Client verlagern – und was müsste trotzdem auf dem Server maßgeblich bleiben?

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.

- ✓ **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."*
- ✓ **Beweisen Sie, dass es wirklich funktioniert.** *"Geh dieses Projekt Ergebnis für Ergebnis anhand der Modulspezifikation unter <https://learnwisej.com/assets/courses/drawing-and-painting/specs/module-05.md> durch und zeig mir den Beleg, dass jedes wirklich erledigt ist – den Codepfad, worauf ich klicken muss und was ich sehen sollte. Wo ein Ergebnis nur teilweise erfüllt ist, sag das klar, statt es schönzureden. Ergänze dann, was fehlt."*



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.