

未経験のフレームワークでも、 AIと一緒に実装できた話

Kagu3

自己紹介

- バックエンドエンジニア
- 主にC#を使用
 - ASP.NET
 - Unity

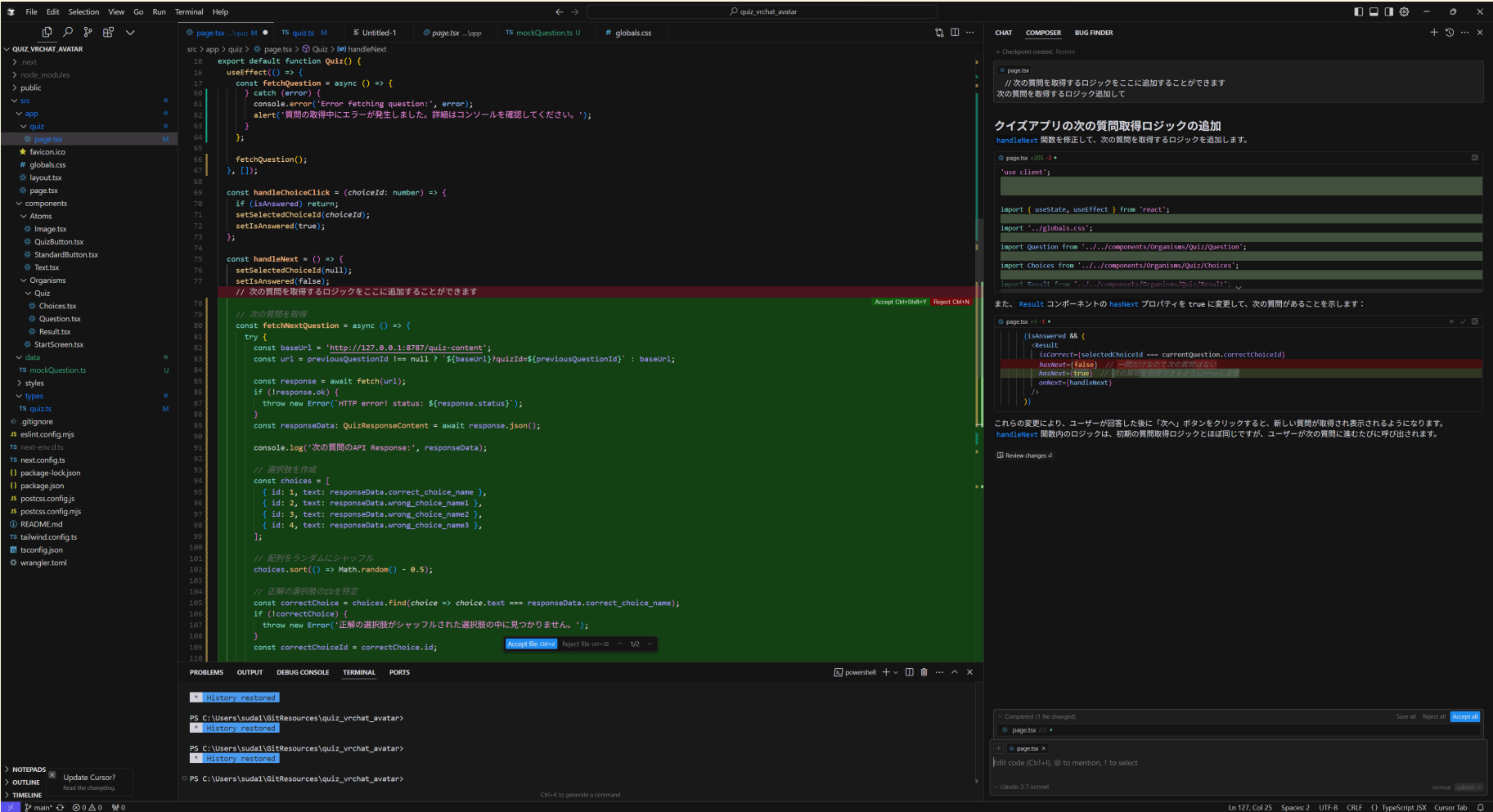
前提

- Cursorの使用感についてのLTです
- MCPについては今回触れていません

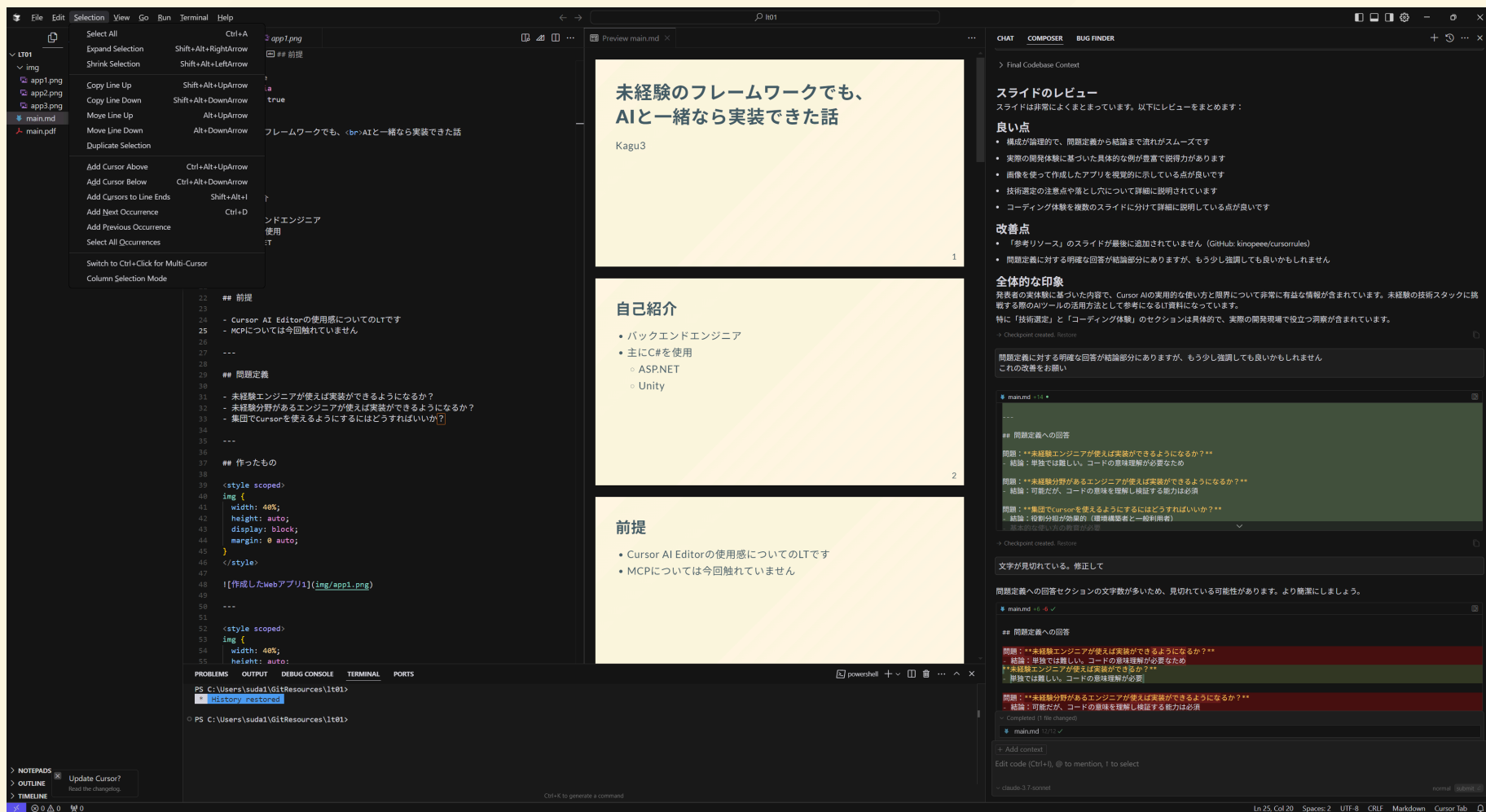
Cursor とは

- AIを搭載したコードエディタ
- ChatGPTのような対話型インターフェースを内蔵
- コードの生成・修正・バグ修正をAIが支援
- プロジェクト全体のコードを理解して回答
- コマンド機能（@Doc, @URL, BugFixなど）が便利

Cursorの開発画面



このスライドもCursorで作成しました



問題定義

- 未経験エンジニアが使える実装ができるようになるか？
- 未経験分野があるエンジニアが使える実装ができるようになるか？
- 集団でCursorを使えるようにするにはどうすればいいか？
 - エディタをCursorに変更させるためのコスト

作ったもの

アバター名分かるかな？



Kyoto

Sanyo

Akyo

Ukyo

アバター名分かるかな？



Kyoto

Sanyo

Akyo

Ukyo

正解！

アバター名分かるかな？



Sanyo

Kyoto

Ukyo

Akyo

不正解...

今回使用した技術スタック

ほぼすべてが**未経験**の技術でした！

- Cursor（未経験）
- TypeScript（ほぼ未経験）
- Next.js（未経験）
- Hono（未経験）
- Cloudflare各種サービス（すべて未経験）
 - Pages, Workers, R2, D1

要求定義・要件定義

- 要求を書いて、AIに清書してもらう
- 要求定義から要件定義を提案してもらい、修正
- PlantUMLの図も自動生成してくれる

技術選定

- 技術選定は相談できるが、最終決定は自分ですることになる
- 知らない技術は選択肢に出てこない
 - Honoを知らなかったので選択肢に出なかった
 - 偶然、別の場所でHonoの存在を知った
- 選択肢を広げる聞き方をしないと、選択肢を絞っていく形に収束しがち
 - "〇〇と△△どちらがいい?" → "〇〇がいいです"
 - "他に選択肢は?" と聞く必要がある

環境構築での体験

- 全自動ではないが、ChatGPTより便利
 - コードを見ながら回答してくれる
 - 修正はApplyで簡単に適用可能
 - ターミナルコマンドも提案・実行してくれる
- 公式ドキュメントのURLを参照させると効率アップ

コーディング体験 (1)

- 要件定義をもとに実装してもらう
- 一見良い実装だが...
 - page.tsxに全コードが集中（コンポーネント分割なし）
 - DIVタグだらけ（WAI-ARIAガイドライン無視）
- モックとしては良いが、保守性に難あり

コーディング体験 (2)

- バグ修正能力
 - BugFixコマンドで多くは解決
 - エラーログを読ませて解決可能
 - 環境構築の問題は解決に時間がかかることも
- インフラ問題は苦手
 - ネットワーク環境などの情報がないと一般的な解決策しか提示できない

コーディング体験 (3)

- 環境の違いによる問題
 - HonoフレームワークでCloudflare Workersを実装中に、Node.js標準ライブラリを使用（当然エラーに）
 - Windows環境なのにBashコマンドを提案してくる
- 有名な実装例に引っ張られる傾向
 - 技術選定基準に"一般的かどうか"の視点が必要
 - 特殊な環境やフレームワークの制約を理解していない場合がある

結論：Cursor AIの効果的な使い方

- プロジェクトフォルダ内に存在しない知識は自動取得されない
- ドキュメントは重要（テキスト形式で作成すべき）
- Ruleの記載は必須
- @コマンド（@Doc、@URLなど）も活用する
- 技術選定や開発環境の選択に、"一般的かどうか"の視点が必要

問題定義への回答

未経験エンジニアが使えば実装ができるか？

- 単独では難しい。先にコードの意味理解が必要

未経験分野があるエンジニアが実装できるか？

- 可能だが、コードの内容を学習しつつ実装する必要がある

集団でCursorを使うには？

- 役割分担（環境構築者と一般利用者）
- 一般利用者には、基本的な使い方の教育が必要