

AI との対話によるプログラミング実践

— 一般公開展示用 3D 分子パズルゲームの開発 —

○鈴木 和磨
分子科学研究所 技術推進部

1. 開発の背景と目的

分子科学研究所の一般公開では、一般の方に科学の面白さを伝えるインタラクティブな展示を用意している。本プロジェクトでは、一般公開向けの 3D 分子パズルゲームを AI との対話を通じて開発した。Python や 3D グラフィックスの専門知識が限られている状態から、AI の支援を受けながら 2 ヶ月の期間内に完成させた。

開発したゲームは、分子式から構造異性体（同じ分子式で結合の仕方が異なる分子）を作り出すパズルである。3D 空間に配置された原子をクリックして結合を作る直感的な操作性により、化学知識がない来場者でも楽しめるよう設計した。

2. ゲームの基本設計と技術構成

ゲーム画面は左側パネル（スコア表示）、中央の 3D 分子操作エリア、右側パネル（操作ボタン）で構成（図 1）される。制限時間は 2 分間で、新しい構造発見ごとに 30 秒追加される。

実装には、Python、PyQt5（UI）、PyOpenGL（3D 表示）、RDKit（分子構造処理）、MariaDB（データベース）を使用した。

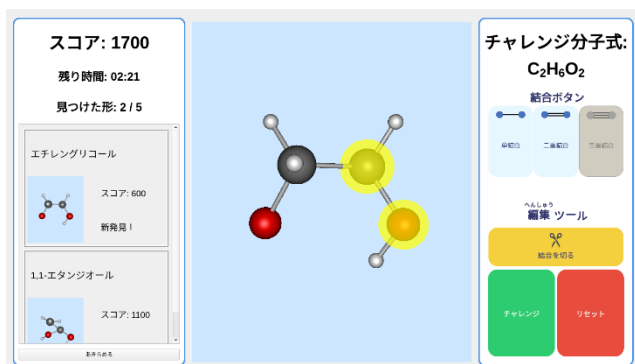


図 1 実際のゲーム画面

3. AI との協働開発プロセスと方法論

人間側は創造的アイデアの提案、要件定義、品質評価を担当し、AI は技術実装、コード生成、デバッ

グ支援を担当することで効率的な開発を実現した。開発には Claude 3.5 Sonnet と v0 の 2 つの AI システムを使用した。Claude はほぼ全ての作業に、V0 は UI デザイン生成に活用した。

開発では段階的アプローチを採用し、基本的機能のプロトタイプから徐々に機能を拡張した。最初は、AI に「Python で Avogadro (分子可視化プログラム) みたいなプログラムは作れますか？」と質問から始め、PyOpenGL による 3D 分子表示機能の実装コードを得た（図 2）。

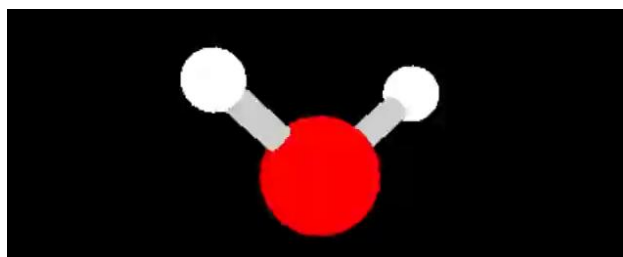


図 2 3D 分子表示のプロトタイプ

複雑な機能開発では、事前にモジュール設計を AI に提示することで効果的な協働が可能になった。例えば、構造異性体判別モジュールのエラーを相談すると、関連する別のモジュールについての的確な修正を提案してきた。

UI デザインでは v0 にデザイン案を生成させ、それを Claude に PyQt5 実装コードとして変換させることで、デザイン専門知識がなくても見栄えの良い UI を実現した（図 3）。

4. 開発過程での技術的課題と解決方法

4.1 コード管理とリファクタリング

AI を活用した開発では、コード品質維持のためリファクタリングが重要。AI は機能するコードを生成するが最適化は自発的に提案しないため、定期的に分割・整理を明示的に依頼した。例えば 3D 分子表示モジュールを機能ごとに分割し可読性を向上させた。また、モジュールサイズを適切に保つことで AI

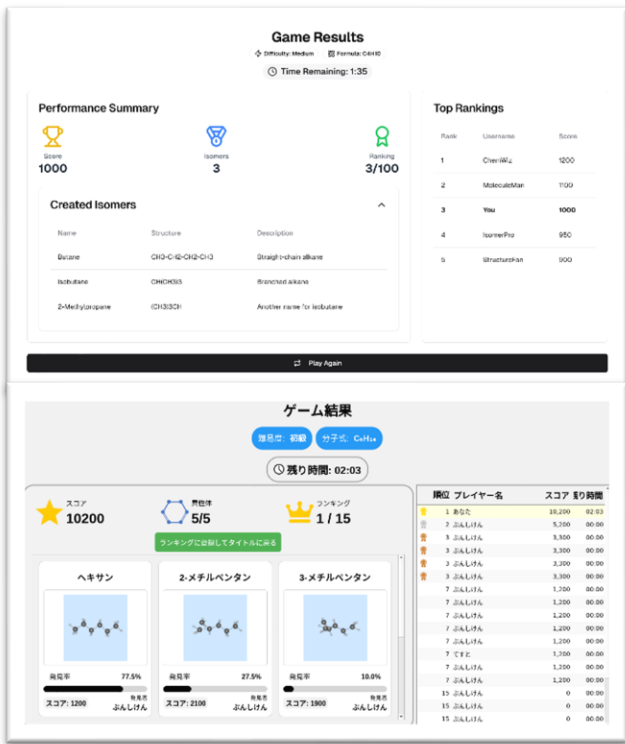


図3 v0 提案デザイン(上)と実際のUI(下)

のエラー率を低減し、命名の不統一やコード重複などの問題は定期的なリファクタリングで解決した。

4.2 技術的課題とデバッグ支援

開発過程で困難だったのは 3D 空間での原子操作問題である。AI に解決策を依頼しても完全解決に至らない場合、「ロジックの流れを追跡するデバッグメッセージを追加して」と依頼し座標変換や関数呼び出しの流れを可視化し、問題の本質を特定できた。

データベース設計においては、分子データやプレイヤー情報の JSON 構造を AI に提示し、適切なデータベース設計を依頼した。AI は主キーや外部キーを含む設計を提案し、自発的に SQL 生成や接続コードも提供してきた。これにより、データベース実装の時間を大幅に短縮した。

4.3 AI の制約への対応と効果的な協働

AI サービスの処理文字量には制限があるため、「引継ぎ文」を AI に生成させる手法が効果的だった。「引継ぎ文」とは、セッション終了時に開発内容・課題・次の作業を要約してもらった文章で、プロジェクト目的、実装済み機能、進捗状況、技術課題と解決策、次回の優先事項を含む。この文章を新セッション開始時に提示することで、開発の連続性を保ちながら長時間開発を続けることができた。

また、モジュール間の関係性を示す設計図をテキ

ストで作成し定期的に共有した。クラス構造やデータフロー、責任範囲を明示することで、AI がシステム全体を理解した適切なコードを生成できるようになった。

マルチモーダル入力の活用も有効で、v0 生成の UI モックアップを Claude に渡して PyQt 実装コードを得るなど、視覚的要件の実装が容易になった。AI の限界に対しては、同じ課題に 2 回質問しても解決策が得られない場合は人間主導で解決するルールを設けた。

5. AI との協働開発から得られた知見と応用可能性

AI 協働開発の経験は、情報技術職員の日常業務にも広く応用できる。マニュアル作成、データ可視化、複雑なコード解析、プレゼンテーション資料作成やトラブルシューティングなど、様々な業務で時間短縮や効率化を実現できる。私の場合、例えばスパコン利用者向けガイドライン作成や都道府県別利用者数データの視覚化などが容易になった。

6. プロジェクトのまとめと今後の展望

本プロジェクトでは、AI との協働により、2 ヶ月の短期間で 3D 分子パズルゲームを開発し、約 1,700 名が来場した一般公開イベントで展示した。成功の鍵は、人間側が創造的アイデアと要件定義を担当し、AI が技術実装とコード生成を担当する明確な役割分担と段階的開発アプローチにあった。

次回の一般公開に向けては、UI の改善、ゲーム性向上を計画している。

本プロジェクトを通して、「専門分野の壁を越えてチャレンジできる」AI の強みは、多様な技術課題に対応する必要がある技術職員にとって極めて価値があると実感した。

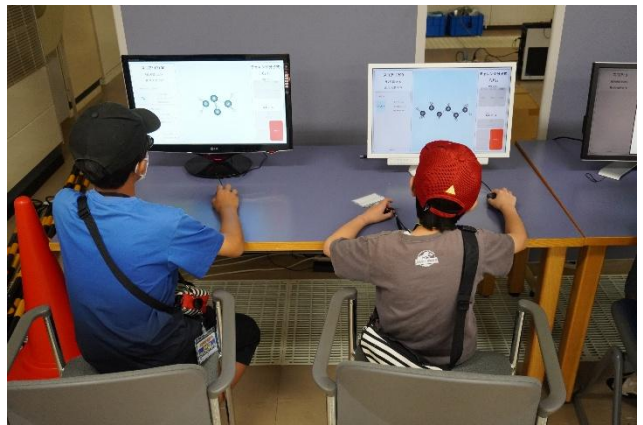


図4 ゲームで遊ぶ子供たち