

1. 担当 PM

竹迫 良範（神山まると高専 デザイン・エンジニアリング学科 教授）

2. クリエータ氏名

関口 祐豊（明治大学 大学院博士後期課程 1 年）

3. 委託金支払額

2,603,500 円

4. テーマ名

AI と学習者と教育者の協調によるプログラミング課題の採点システム

5. 関連 Web サイト

なし

6. テーマ概要

本プロジェクトでは、プログラミング課題に対する LLM によるコードの自動評価と、学習者同士のクラウドソーシング型の相互評価を組み合わせた新しい採点システム「PP-Checker」を開発した。従来の自動採点システムは、標準出力を基にした正誤判定が主流であり、視覚的・動的なプログラムの採点には対応しきれていなかった。また、Processing のような課題では、画面の見た目や動き、入力に対する反応を実際に観察しながら評価する必要があるため、教員や TA が提出物一つずつ確認する負担が大きくなりやすかった。さらに、大規模講義では提出が集中し、採点待ちやフィードバックの遅延が発生しやすく、学習の停滞に繋がっていた。これらの課題を解決する採点システムを開発し、実際の大学生向けのプログラミングの授業で活用し、フィードバックを受けながら実用的なプロダクトとなるよう改善を繰り返した。

7. 採択理由

本提案は教育現場の実態を丁寧に踏まえており、特に TA 不足や講義の大規模化によって発生する採点の負担、そしてフィードバックの遅延による学習者のモチベーション低下など、現場の課題に直結する解決策をシステムと仕組み

の両面から解決しようとしている点を評価した。特に、学習者同士による相互評価を単なる分担作業としてではなく、「他者の提出物を評価するプロセス自体を学びの一部」として位置づけている点は、42Tokyo など IT エンジニア養成機関でも実際使われているピアレビューの仕組みと似ており、大変実践的である。

また、本提案のベースになっている既存の採点システムは 100 名規模の大学のプログラミング演習の講義において運用実績があり、提案内容が単なるアイデア段階を超えて、実証済みのプロトタイプである点も評価できる。

実際に出題するプログラミングの演習課題についても、正解のあるアルゴリズムのソースコードを厳密に採点する方式以外にも、従来対応が難しかったクリエイティブな要素、動的・視覚的な課題に対しても柔軟に採点が可能な新たな評価基盤を構築しようとしていることも高く評価できる。

API ベースで設計し、Web ベースで簡単にインテグレーションができるのであれば、大学の授業のみならず中学・高校・高専の授業、またオンライン教育サービス、企業の研修など、さまざまな教育現場への水平展開が期待できる。単なる研究開発にとどまらず、さらなる社会実装へ向けた展開を期待している。

8. 開発目標

Processing のような視覚的なアニメーションの動きがあったり、アクションに応じた動的な変化があったりするプログラミング課題において、教員・TA の採点負担を軽減しながら、学習者に対して迅速にフィードバックを返せる採点支援を実現することを目標とした。正解や数値目標のある競技プログラミングのような自動採点課題と異なり、標準出力による正誤判定に依存しない評価基盤を構築し、画面の見た目や動きとして現れる正しさを扱えるようにすることで、従来の自動採点では対応が難しかった課題にも対応することを狙った。また、提出直後に AI が修正の起点を提示し、学習者同士の相互評価によって多様な視点からのコメントを得て、さらに教育者が最終的な判断を行う役割分担を設計することで、採点効率と学習効果の両立を目指した。

9. 進捗概要

視覚的・動的なプログラミング課題に特化した Web ベースの採点プラットフォーム PP-Checker を開発した。学生から見た課題提出画面の例を図 1 に示す。

(1) LLM による一次評価、学習者同士の相互評価、教育者による最終確認の 3 つのフェーズを連携させ、提出から再提出までの循環を短時間で回せる構成を実装した。学習者向けには、課題一覧、課題提出、AI フィードバック表示、相互評価の各画面を統合し、ブラウザ上でコードの記述と実行結果の確認、

フィードバックの参照と修正を一体的に行えるようにした。これにより、フィードバック待ちによる停滞を減らし、試行錯誤を継続しやすい環境を実現した。

- (2) 相互評価では、他者の提出物を実行して観察し、ルーブリックに沿って評価とコメントを返す仕組みを実装した。評価結果は提出者に返され、修正と再提出へと繋がられるようにした。教育者向けには、提出物一覧画面と採点画面を整備し、提出状況や確認状況を一覧で把握しながら、コードと実行結果を一画面で確認できるようにした。LLM の評価プロンプトやパラメータを運用中に調整できる仕組みや、判断理由を確認できるチャット機能も実装した。
- (3) Processing.js を用いたブラウザ実行基盤を整備し、互換性の差を補う前処理や互換レイヤを実装した。画像や音声を含む課題への対応、リソース管理やパス変換、音声再生制御も整備した。さらに、Arduino 等のフィジカルコンピューティング課題に対応するための動画提出機能も実装し、実世界の挙動も評価対象として扱えるようにした。成績の CSV 出力機能や多言語対応も整備した。
- (4) 100 名規模の大学のプログラミングの授業で実際にこの採点システムを運用し、400 名以上のユーザに利用された実績を得た。PP-Checker の導入により、学習者がフィードバックを受けてから再提出までの平均時間は約 31 分から約 12 分へ、課題達成までの平均時間は約 59 分から約 48 分へ短縮できたことを実際の採点業務の中で確認できた。さらに、SUS スコアは 76.4 であり、平均値を上回るユーザビリティを実現できた。
- (5) 教育機関向けの PP-Checker に加え、一般学習者向けの Open PP-Checker も整備し、授業内にとどまらないオンライン学習基盤として展開できるようにした。

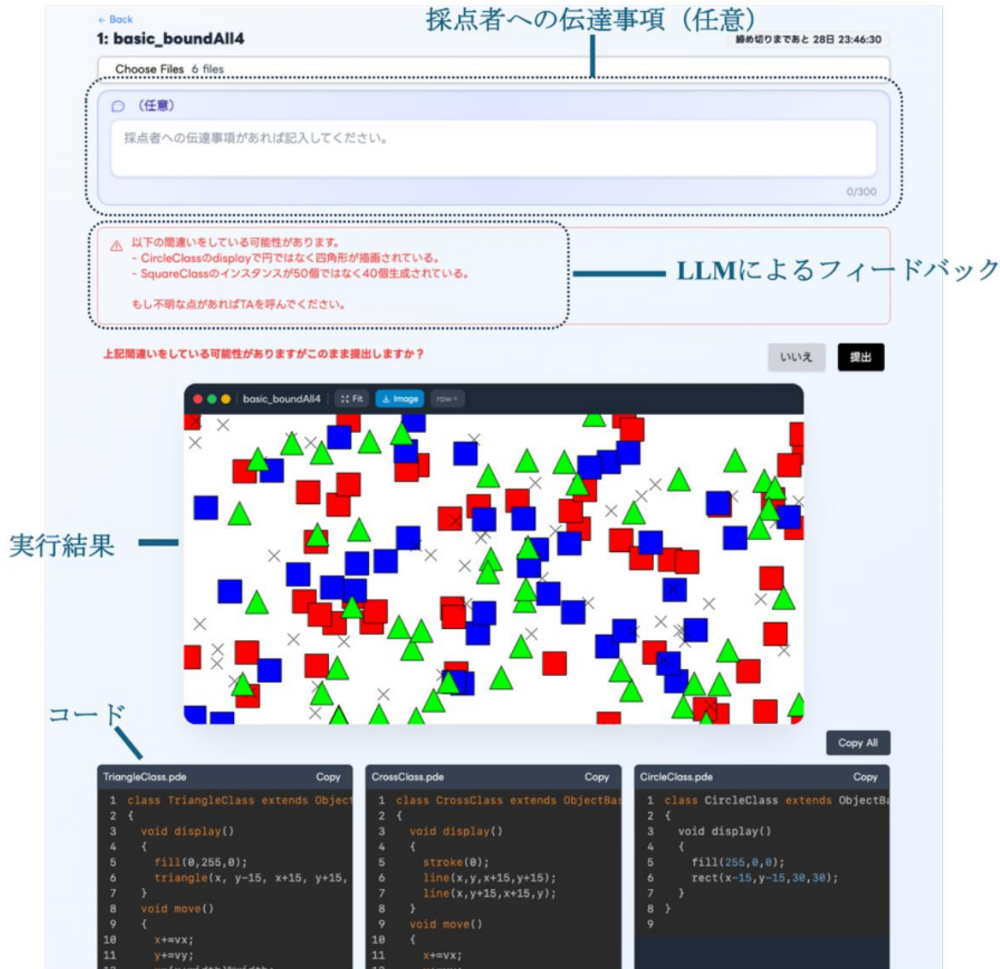


図 1 : PP-Checker の課題提出画面

10. プロジェクト評価

本プロジェクトは、机上のアイデアに留まらず、実際の現場に耐え得るプロダクトとして泥臭いことも含めてきちんと仕上げ、実際の授業運用の中で回る形にまでシステムを作りこんだ点を高く評価した。難易度の高い実装もあった中で、シンプルで細かい機能ではあるものの、レビュー完了時のメール通知の機能など、採点業務プロセスを止めずに回し続けるために必要な運用上の勘所をしっかり実装していることに好感を持った。実際に 400 人以上のユーザが利用し、累計 28,568 件の提出物を処理していることから、単なるプロトタイプの試作や PoC（概念実証）に留まらず、実際の教育現場で機能する採点支援基盤システムとして成立していることが分かる。

11. 今後の課題

現在のシステムは Processing を前提に設計されているため、他プログラミング言語への拡張が一つの重要な課題である。視覚表現やインタラクションを含む課題は他のプログラミング言語の JavaScript や Python、Java などでも存在

しうることから、そうした課題にも AI、学習者、教育者の協調による採点支援を適用できれば、本プロジェクトの意義はさらに広がる。プログラミング言語ごとに実行環境のレイヤやライブラリ、入出力形式が異なるため、実行基盤を一般化することは簡単ではないが、クリエイターには Processing.js との互換性ギャップに対して前処理や互換レイヤを実装した取り組みや、動画提出機能を実装した経験があるため、今後継続的に取り組んでいけば解決できる課題だと信じている。今後は、プログラミング言語ごとの実行環境をテンプレート化し、課題種別に応じて柔軟に切り替えられるような設計へ発展させることが課題である。