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
― 早い評価と深い学びを提供するプログラミング教育環境の実現 ―

1. 背景

プログラミング学習では，提出後に適切なフィードバックを受け，修正を重ねることが重要である．特に初学者にとっては，早い段階でのフィードバックが行われるかどうかが，学習の継続や理解の深まりに大きく影響する．

一方で，Processing のような視覚的・動的プログラミング課題では，正しさが標準出力の一致だけでは判断できず，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．そのため，既存のテストケースベースの自動採点をそのまま適用することが難しく，教員や TA が提出物を一つずつ開いて確認する負担が大きくなりやすい．

また，大規模講義では，提出が集中することで採点待ちが発生しやすく，フィードバックの返却が遅れやすい．フィードバックが遅れると，学習者はどこを直せばよいか分からないまま作業を止めたり，別の課題へ進んでしまったりする．このように，視覚的・動的プログラミング課題では，評価の難しさがそのまま学習の停滞に繋がりやすいという課題がある．

2. 目的

本プロジェクトの目的は，視覚的・動的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である．Processing のように，正しさが画面の見た目や動きとして現れる課題では，既存のテストケースベースの自動採点をそのまま適用することが難しく，人手による確認に大きな負担がかかっていた．そこで，採点効率を高めつつ，提出→フィードバック→修正→再提出の循環を止めない仕組みを構築することを目指した．そのために，本プロジェクトでは，AIによる即時フィードバック，学習者同士の相互評価，教育者による最終確認を組み合わせた協調型の採点支援を実現する．AI は提出直後に修正の起点を返し，学習者同士の相互評価は多様な視点からのコメントを生み，教育者は最終的な合否判断と講義としてのフィードバックを担う．この役割分担により，採点の効率を高めるだけでなく，評価を受けることや他者を評価すること自体が学びに繋がる環境を作ることを目指す．

さらに，本プロジェクトでは，教育機関向けの採点支援システムとしての PP-Checker と，一般学習者向けオンライン学習サービスとしての Open PP-Checker の両方を視野に入れ，講義内の運用と自律的な学習の双方で活用できる基盤へ発展させる．

3. 開発の内容

本プロジェクトは，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．PP-Checker は大きく分けて以下の 3 つのフェーズで機能する．

① LLM による一次評価：学生が課題を提出すると，LLM が即座にコードを静的解析し，教育的配慮に基づいた（直接答えを教えない）フィードバックを行う．
② 学習者同士の相互評価：LLM の評価を通過した提出物に対し，他の学習者がブラウザ

上で実行・操作し，コードを閲覧することなく実行結果のみを確認することでフィードバックを行う．

③ 教育者による最終確認：教員や TA が採点状況をリアルタイムで把握できるインタフェースから，効率的にフィードバックと成績づけを行う．

各フェーズを連携させることで，提出→フィードバック→修正→再提出の循環を短い時間で回せるようにした点が，本システムの大きな特徴である．従来のように教員や TA が全提出物を逐次確認するのではなく，まず LLM が修正の起点を返し，次に学習者同士の相互評価で多様な視点からのコメントを与え，最後に教育者が最終判断を行う構成とすることで，採点効率の向上と学習機会の拡張を両立させた．

学習者向けには，課題一覧画面，課題提出画面，AI フィードバック表示機能，相互評価画面を実装した．課題一覧画面では，取り組むべき課題や進捗状況，レビュー結果を確認できる．課題提出画面（図 1）では，学習者がブラウザ上でコードを書き，実行結果を確認しながら提出できるようにした．また，AI フィードバック機能を同一画面に統合することで，学習者は提出後すぐに指摘内容を確認し，そのまま修正へ移れる．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くした．

図 1. 課題提出画面

相互評価機能では，評価者が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せるようにした（図 2）．視覚的・動的課題では，コードそのものだけでなく，画面の見た目や動き，入力に対する反応を観察しながら判断することが重要になるため，相互評価は提出者に多様な視点からのフィードバックを返す手段として有効である．同時に，評価者にとっても，他者の成果物を見ながら仕様を捉え直し，自分の実装を相対化して

見直す学習機会となる．さらに，返ってきたレビュー結果を学習者が確認できるようにし，フィードバックを修正と再提出へ繋げやすくした．

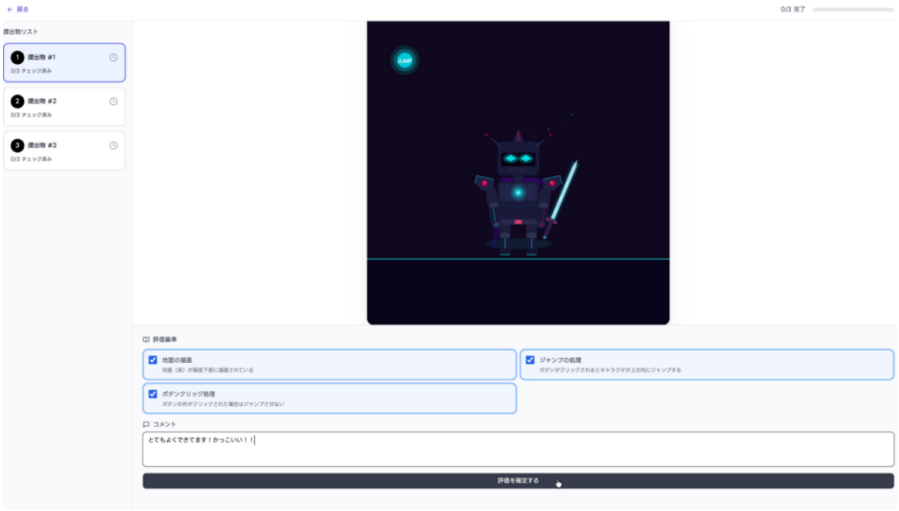

図 2. 相互採点画面

教育者向けには，提出物一覧画面と課題採点画面を実装した．提出物一覧画面では，提出者，課題，提出時刻，最新提出かどうか，確認済みかどうかなどを一覧で把握できる．これにより，複数の TA が関わる講義でも重複確認を避けやすくした．課題採点画面（図 3）では，コード，実行結果を一画面に集約し，教員や TA が最終確認を行いやすくした．また，LLM 評価に用いるプロンプトの修正やパラメータ変更も同じ画面から行えるようにし，実運用の中で評価精度を改善できるようにした．さらに，不慣れな TA を支援するため，LLM がなぜその判断を行ったのかを質問できるチャット機能も実装した．

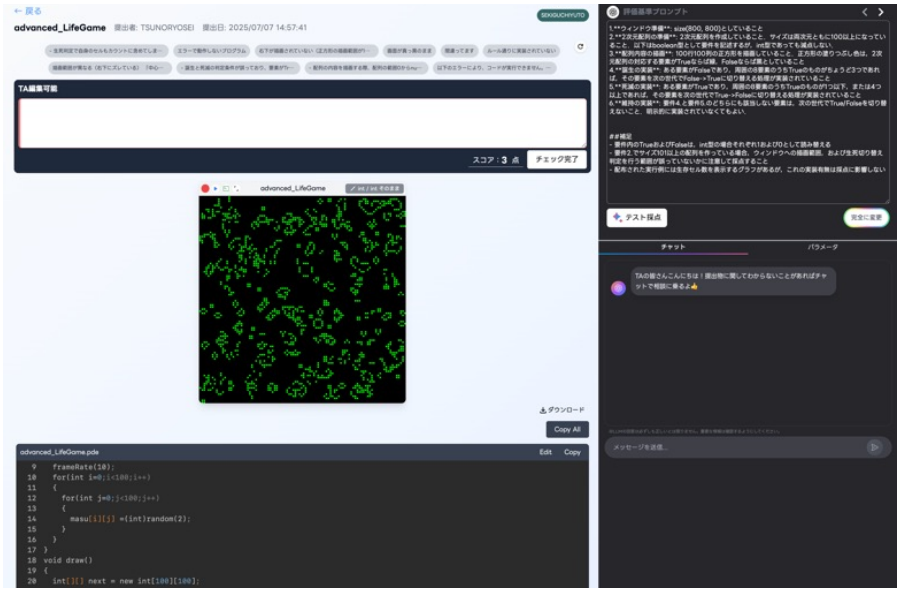

図 3. 課題採点画面

視覚的・動的課題を扱うための実行基盤についても工夫を行った．ブラウザ上で提出物を実行するために Processing.js を利用したが，Processing 環境との間には互換性の差があるため，未対応関数や構文を補う前処理や互換レイヤを実装した．また，画像や音声を

含む課題に対応するため，提出フォルダ内のリソース管理，パス変換，音声再生制御も整備した．さらに，Arduino 等を用いたフィジカルコンピューティング課題にも対応するため，動画提出機能を追加し，画面上だけでなく実世界の挙動も評価対象として扱えるようにした．加えて，講義運用を支えるため最終的な成績を CSV 形式で出力する成績ダウンロード機能，日本語，英語，中国語，韓国語に対応した言語切り替え機能も整備した．

さらに，本プロジェクトでは，教育機関向けの PP-Checker に加えて，その考え方を一般学習者向けに再構成したオンライン学習プラットフォーム Open PP-Checker も整備した（図 4）．PP-Checker では講義内の採点支援を目的として，AI による一次評価，学習者同士の相互評価，教員・TA による最終確認を組み合わせていたのに対し，Open PP-Checker では，AI による即時フィードバックと学習者同士の相互評価を中心に，学習者が自律的に修正と学習を重ねられる環境として設計している．これにより，本プロジェクトの成果を教育機関内の授業支援に留めず，一般向けの学習基盤へ広げる可能性を示した．

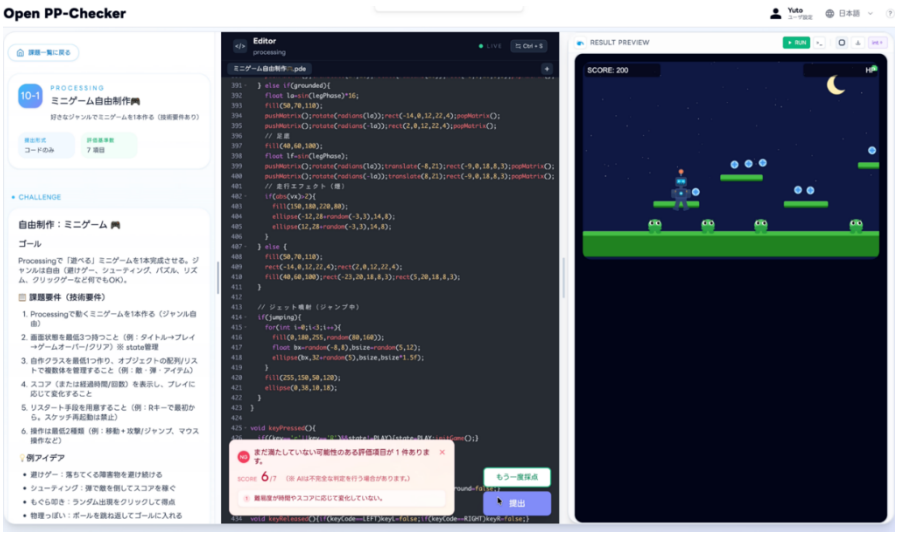

図 4. Open PP-Checker

4. 従来の技術との相違

従来のプログラミング課題支援では，LMS 等による提出管理や，テストケースベースの自動採点が広く利用されている．これらは，課題の提出・回収や，標準出力が明確に定まる問題の正誤判定には有効である．一方で，Processing のような視覚的・動的プログラミング課題では，正しさが画面の見た目や動き，入力に対する反応として現れるため，テストケースだけで十分に評価することが難しい．その結果，教員や TA が提出物を一つずつ実行して確認する必要があり，大規模講義では採点待ちやフィードバック遅延が発生しやすかった．また，近年は LLM を用いたフィードバック支援も注目されているが，LLM 単体では，誤判定や曖昧な指摘が避けられず，視覚的・動的課題のようにコードだけから判断しきれない要素を多く含む場合には，それだけで採点を完結させることは難しい．

これに対して本プロジェクトでは，AI による即時フィードバック，学習者同士の相互評価，教員・TA による最終確認を一つの採点フローとして統合した．AI は提出直後に修正の起点を返し，相互評価は実行結果の観察に基づく多様なフィードバックを生み，教員・TA は最終的な合否判断と講義としてのコメントを担う．このように役割を分担することで，AI だけでも，

人手だけでも実現しにくかった視覚的・動的なプログラムの採点効率と学習機会の両立を可能にした点が，従来技術との大きな相違である．さらに，本プロジェクトでは，相互評価を単なる補助機能ではなく，評価者自身の学びにも繋がる仕組みとして位置づけた．他者の成果物を観察し，評価し，コメントとして言語化する過程を通じて，評価者は仕様理解を深め，自分の実装を相対化して見直す視点を得ることができる．本成果は，採点を効率化するだけでなく，評価の過程そのものを学習機会へ変えている点でも，従来の技術と異なる．

5.　期待される効果

本プロジェクトによって期待される効果は，学習者，教員・TA，授業運用全体の 3 つの観点に整理できる．まず学習者に対しては，提出後すぐにフィードバックを受けられることで，修正のタイミングを逃しにくくなることが期待される．LLMによる一次評価により，どこを見直せばよいかを早い段階で把握しやすくなり，相互評価によって自分だけでは気づきにくい改善点にも気づける．その結果，提出→修正→再提出の循環が短い時間で回り，学習の停滞が減ることが期待される．また，相互評価は提出者へのフィードバック手段であるだけでなく，評価者にとっても，他者の成果物を見ながら仕様を捉え直し，自分の実装を相対化して見直す学習機会になる．教員・TA に対しては，採点負担を軽減し，より教育的な支援へ時間を振り向けやすくする効果が期待される．提出物一覧画面や課題採点画面に，コード，実行結果，AI フィードバック，相互評価結果などを集約することで，確認のための手間を減らしやすい．さらに，AI による一次評価や学習者同士の相互評価を前段に置くことで，教員・TA は最終確認と講義としてのフィードバックに集中しやすくなる．授業運用全体としては，フィードバックの流れが安定することで，講義の進行そのものが円滑になることが期待される．大規模講義でも，AI と学生間の相互評価を組み合わせることで，教育者だけに負荷を集中させずにフィードバックを回しやすくなる．

6.　普及の見通し

本プロジェクトの成果は，視覚的・動的プログラミング課題を扱う幅広い教育現場に展開できる可能性を持っている．特に，提出数が多く，教員や TA の確認負担が大きくなりやすい授業では，本システムの導入効果が大きいと考えられる．実際に PP-Checker は，明治大学総合数理学部 1 年生の 100 人規模のプログラミング講義等に導入され，すでに運用されており，これまでに 400 人以上のユーザに利用されている．これらの実績は，本システムが試作段階に留まらず，実際の教育現場で利用可能な採点支援基盤として機能していることを示している．また，本プロジェクトでは，教育機関向けの PP-Checker だけでなく，一般学習者向けの Open PP-Checker も実装した．これにより，授業内での利用に留まらず，自律的なプログラミング学習の場にも展開できる見通しがある．今後は，すでに導入済みの講義での運用を継続しつつ，他の教育機関での試験導入や，Open PP-Checker の一般公開を通じて利用者層を広げていく見通しである．

7.　クリエータ名（所属）

関口　祐豊（明治大学　大学院博士後期課程 1 年）