A Demo of AI Feedback in Pyrates: Supporting Students' Transition to Text-Based Programming

Anonymous

No Institute Given

Abstract. Learning programming can be especially challenging for high school students transitioning from block-based to text-based environments. Serious games like Pyrates offer engaging learning opportunities to do so, but students often require personalized support that busy K-12 teachers might struggle to provide in a timely manner. This demo presents our AI-augmented version of Pyrates, featuring an adaptive feedback policy that predicts and delivers formative feedback based on students' in-game actions and code. Deployed in multiple high school classes, this system addresses common difficulties in the game. We also preview how LLMs can further enhance feedback personalization.

Keywords: Serous game · Feedback policy · Learning programming

1 Introduction

Pyrates (`https://py-rates.org/`) [3] is a platformer-style serious game aimed at introducing Python programming to 10th-grade students with prior block-based coding experience. As such, it supports the critical transition from block-based to text-based programming featured in many high school computer science curricula [4]. Players control a pirate avatar by writing Python code to achieve level-specific goals. Each of the levels introduces a novel programming concept (e.g., loops, conditions, variables). Pyrates is meant to be exploratory and open-ended, meaning that the fundamental programming concepts involved in each of the levels are not explicit, but are made necessary by the game problems to be solved, thus aligning with the constructivist learning paradigm [6]. Pyrates is used on a rather large scale, with over 250,000 game sessions played to date.

The core of the demo is the integrated AI-driven adaptive feedback system [1]. This system utilizes machine learning (ML) models trained on student data (N=215), including their in-game actions and code, to predict the feedback type that expert teachers (N=7) would provide in similar situations. The feedback types are based on the well-established Narciss' framework [5] and include: Task rule (CTRL, elicits the level's goal and available control functions); Concept (CONC, elicits the programming concept needed for the level); Procedural indications (IMPL, guides on how to implement the concept in Python); Correction response (SOLU, provides a correct code solution). A formal evaluation study [2] showed that students (N=190) in the experimental group (with the feedback

system) progressed significantly further in the game compared to a control group without feedback, with a difference of 1.11 levels or nearly 25%. Students also expressed positive perceptions of the system, with 77% expressing intent to reuse it.

2 Interactive Demo Description

This interactive demo offers participants a hands-on opportunity to experience how adaptive AI feedback can be integrated into an exploratory serious game to support high school students learning Python. It will highlight both the technical and pedagogical decisions underlying the design of the system. During the demo, we will take on the role of a high school student using Pyrates with the AI-driven adaptive feedback system enabled. The demo involves progressing through several game levels that require applying Python programming concepts. The demonstration will include:

- Student Interaction: Using the Pyrates interface, writing and executing Python code, interacting with game elements, and consulting the programming memo, simulating an authentic student experience.

-Requesting Feedback: Requesting feedback by clicking the dedicated "Help" button, and exploring the data sent to the AI policy in real-time.

- Adaptive Feedback Delivery: Exploring the outputs of the AI policy, including the predicted feedback type (CTRL, CONC, IMPL, or SOLU), delivered by the in-game parrot tutor.

- Work-in-progress with Large Language Models: Experimenting with a novel version of the feedback system based on LLM to render the feedback, and exploration of the prompting strategies. Due to the open-ended nature of the game, the prompts are built to include detailed information about the game level and goals, students' data, and feedback types, for the LLM to generate context-aware feedback suitable to the game level.

- Code and data: Introducing the open-source code of our AI system, as well as the publicly available data collected in our study.

3 Conclusion and Future Work

The Pyrates interactive demo illustrates the feasibility and positive impact of an AI feedback policy that adapts to student code and behavior in an exploratory serious game that supports the transition from block-based to text-based programming. For future work, we aim to enhance the system by improving the predictive accuracy of the AI policy, exploring its fairness across diverse student groups, and formally assessing the added value of LLM-based feedback generation. Our ongoing goal is to evaluate the system's impact on student learning and teacher experience in real classrooms to support the large-scale adoption of Pyrates.

Acknowledgments. This work was supported by XXX.

References

1. Anonmized: Anonmized. In: XXX (2025)
2. Anonmized: Anonmized. In: XXX (2025)
3. Branthôme, M.: Pyrates: A serious game designed to support the transition from block-based to text-based programming. In: European Conference on Technology Enhanced Learning. pp. 31–44. Springer, Cham (2022)
4. Lin, Y., Weintrop, D.: The landscape of block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. Journal of Computer Languages **67**, 101075 (2021)
5. Narciss, S.: Designing and evaluating tutoring feedback strategies for digital learning. Digital Education Review **23**, 7–26 (2013)
6. Sjøberg, S.: Constructivism and learning. International encyclopedia of education **5**, 485–490 (2010)