

Interdisciplinary Lesson Platform with AI Code Generation

Target audience

Our primary audience is for teachers in secondary education, although we believe this tool can be used by teachers at any level. No Computer Science background is required

Expected watch time:

12 minutes

Description:

The PinCS Interdisciplinary Lesson Platform offers educators a new way to create high-quality interactive lessons that include Computer Science. We have integrated Artificial Intelligence into the platform at a key part of the lesson building process, where educators can create engaging programming lessons without being expert programmers themselves.

For educators, the platform allows users to browse and edit existing content or create their own lessons from scratch. The lesson editing experience is powered by a drag-and-drop library that should feel familiar to users who have created lessons with software in the past (e.g. PowerPoint). For Computer Science education, the platform offers an editor that connects with a selection of “plugins,” which define the relationship between the text editor and the output window. For “console” plugins, the text editor acts as a user would expect, with standard input and output capabilities. For “canvas” plugins, the user is given a small API of drawing functions that render points, lines, and shapes in the output window based on parameters. We also offer educators the ability to write code that runs before and after the student’s code. This allows educators to define their own canvas or console environments for users, building upon the API we provide.

Our lesson editor makes use of AI at the aforementioned stage of the creation process. If a teacher has an idea about how they would like to “extend” the functionality of our console or canvas API, they can write out their idea into the lesson maker, which will use an LLM to generate the “before” and “after” code snippets that provide context for the activity. The LLM is fed the API of the desired plugin type (console or canvas), as well as instructions for how to generate extensions in our “before and after” system. The generated code is loaded directly into the lesson editor’s UI, so teachers can test and modify the extension at will.

The platform currently supports this functionality in Javascript and Python.