Literature Review (material for Chapter 2 Background)

**For Parson’s Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses:**

The Parson’s Problems follows the five core principles. To be more specific, the problems should maximize the engagement with puzzle-like game-style questions, constrain the logic, permit common errors by setting distracters, model good code and provide immediate feedback.

Puzzle setting structures: drag-and-drop style, code fragments can include single or multiple lines, and some subsets of these code fragments can constitute the solution. Click the check button to get feedback. Repeat to 100% correct. The problems can be described in text or activity diagrams. And activity diagrams have more advantages than simple text since students need to read and understand the diagram, in other words, they need to understand the solution logic of problems. To some extent, it seems like to add the method Coached Program Planning in this Parsons Puzzles. For distractors, these components can not only be used to point out some common mistakes but also train some good habits, like using the declared constants, identifying the correct keyword, formatting rules, variable names, and even proper indentation.

Implementation: [Hot Potatoes](https://web.uvic.ca/hrd/halfbaked/)

Future work: different problem types: multiple choice, fill-the-gap, and crossword. Increasing the problem pool -> set up questions for each other. It may not be a good idea, since the teachers are more professional and they can set more significant distractors. Besides, the teachers can check frequently mistakes and explain them in the classroom.

Enhancing user feedback -> impossible to predict every potential error -> give an explanation only for the correct version.

Track their long-term progress, showing the number of errors they make for a given topic over time.

Enhanced User Interface: lack of sound, animation, and color. More stylish interface, incorporating attractive graphics and easy navigation between topics. animation rewards for completing puzzles. Like the small rocket in PythonAnywhere.

Enhanced Data Collection: keep detailed records of which items are chosen during each iteration of the puzzles.

**For Two-Dimensional Parson’s Puzzles: The concept, Tools and First Observations.**

2D parson’s puzzles: The vertical dimension is used to order the lines, whereas the horizontal dimension is used to change control flow and code blocks based on indentation as in Python.

Insert a line between two adjacent lines of existing code. (not supported in Hot Potatoes)

Python, a website where teachers can browse existing puzzles, and create collections with several puzzles for their students.

Different types of puzzles: extra lines (group the distractors with the correct alternative and always keep them next to each other in the initial random order -> reduce the cognitive load), User-created blocks (insert curly braces or indent the code), Context (provides a fixed code around the code to be sorted. It allows larger, and often more concrete, examples to be shown to students)

Tools: ViLLE for Java, CORT

Mark the correct/incorrect position in different colors