Problem Pool

Test Pool

Interface for instructors:

My tests

My questions

Choose a test from the pool. According to:

1. Tags – difficulty or topics
2. Used times(frequency): used to evaluate the quality of the puzzle/ the instructor who created it can get the credits
3. Review(Rating star): rating star or comments from both the instructor and the students

Create a new test

Choose a problem from the pool. According to:

1. Tags – difficulty or topics
2. Used times(frequency)
3. Review(Rating star)

Or create a new problem

Create a problem(instructor)

1. make some syntax rule for the instructors to create a new problem in a coding block(with the line number of the codes):

#d#: distractor

Example: #d# a = 2;

todo: automatically generated

#fs# #fe#: blank to be filled

Example: a = #fs# 2 #fe#

Display: a = \_\_\_\_ (the answer is 2)

#t#: indentation

Example: #ttt# 3 indentation

#c#: comment line

1. set the expected actions of students to solve drag and drop(total times of actions)
2. write the hints(feedback): tips/ learning objectives of the puzzle/ activity diagrams/ flow chart  
   set the max try times before showing the hints
3. set the time limit or not
4. the instructor tags the difficulty and topic of the problem  
   options for difficulty: Easy, Medium, Hard  
   options for topics: choose from default options or create a new topic
5. generate a puzzle according to the coding instructions

Check the grades of a problem

Overall contributes(all students):

1. The number of students who take the test/ solve the problem
2. The normal distribution of time spent
3. The normal distribution of try times
4. The occurrence times of hints info for a student doing the puzzle
5. The pass ratio of the first try/second/ ith try
6. Reviews: rating star or comments

Interface for students:

Track the progress of a specific students

(time spent, try times, passing ratio, type of questions they’re doing correctly)

Dynamically adaptive parsons puzzle

Interface for admin:

Tags of problem

Difficulty:

Length of the code lines

Increasing number of distractors

Unpaired distractors are harder

Hybrid of reorder/indentation/fill in the blank makes it harder

Specific topics

Hybrid of multiple topics

Tag Difficulty calculation: tag from the instructor + students’ time spent/movement/try times/pass ratio data(after students submit the answer).

Tagged by instructors

Topic:

Review:

Used time:

Data analysis:

1. students wrong answer pattern(wrong order or answer)

Problem pool

Feedback:

Programming is about the whole big picture. Each line of codes effects the other. It’s a system. Simply tell me which line is wrong is not helpful cuz I still need to think why it’s wrong.

For example, two dimension array, we usually use nested for-loop to iterate it. If I am given a puzzle to reorder or organize the for-loop, I probably would make a mistake. But if I only get a feedback of which line incurs the error, it’s not helpful. We probably should provide a 2D diagram of how does 2-dimension array look alike, help the students generate a picture in their head

i0 i1 i2 i3

j0: 1 2 3 4

j1: 2 3 4 5

j2: 4 5 6 7

hints: activity diagram

Feedback: correct solution. The learning outcomes of this puzzle.

Content

Difficulty tag:

A panel to show the student’s solution

Use of tools/languages/applications

Partial recorder

Using fragments instead of lines (variable declarations)

it’s hard to build a frame to solve everything. Each puzzle should be designed well. And of course it can be repeatedly used.

Every instructor can code, we should provide a programming window to let the instructor write the specific feedback of a puzzle. Then integrate the code into the system. – formulate some function, like remember the number of feedback info occurrence

The occurrence of feedback info for a student doing the puzzle – measure of the students also the difficulty

Set time limit to ensure that the students use the time on the puzzle. This way, time spent on a puzzle can be a measure of difficulty

Concern: creativity development of students. People have different approach to the task.

Contributor of a puzzle – credits of an instructor

1. build a parsons puzzle library

2. key concepts/learning outcomes/learning objects

algorithm – DP, tree, calculate the combination, nested for-loop, recursive

sorting algorithm

data structure

the order of functions(c++),

the syntax of using pointer(c++)

order to build tcp connection(packet)

object – java (ask students to write class based on the needs of realizing some functions, interface) – fill in the blank

comment style

To build this library, we need to do survey among cs area and other subjects.

It can also be used in programming study for children. (with the pic, video, media function)