Tutor 1

Audiodatei

recording-file (2).m4a

Field Code Changed

Transcript

Tutor 1

Oh, there you are.

Viviane Rehor

I think it's the second one, yeah.

Tutor 1

Hey, OK.

Viviane Rehor

Yeah, actually I don't really know how much explaining I should do, but basically. These are the exercises with the titles and it's your job will be to. Build a programme out of blocks instead of lines that yeah, achieves this and you have an.

Tutor 1

Would like me to.

Viviane Rehor

Explanation with the exercise. Yeah, yeah, yeah.

Tutor 1

Think aloud while I'm doing it or. OK, good. Right. OK, lots of things to work with. That programme takes numbers of a list and. Firstly, outputs the numbers even odd afterwards, whether it is the current maximum OK. Figuring out my. Interface a little bit, so I've got blocks I've got. So this is just. This is the programme, but in Python K. And I want to achieve that. With my blocks right? Just to test, I'm just going to. Where do I construct my solution?

Commented [RV1]: Clue for later dividing the workspace into two halfs

Viviane Rehor

Just answer questions, OK.

Tutor 1

I suppose, yeah.

Viviane Rehor

Yeah, it's basically that's all your works, faith.

Tutor 1

OK, so if I don't need something for or will I need everything? Maybe I'll need everything. OK, let me let me figure out, that's fine. OK. I just want to get a feel for what the blocks I've got. I've got numbers, odd numbers even. I'm just gonna put these all in the place where I can keep track of them. This isn't where I'm actually going to build things, and the current Max that's going to be at some point after them, so just pop that somewhere for now. OK. For each item. In the list. OK, so I'm going to want to do a thing there. That's all good now takes numbers of the list. Firstly outputs

Commented [RV2]: Clue for later adding that information to the tips within odd and afterwards and it should do that for each number, I assume so. For each item in the list. I'm assuming that these are all fine to have here, so if it's even, say the numbers even else. It will be odd. I'm pretty happy with that. UM, I'm going to need to set the Max at some point. Well, for now. I think pop that in if NUM is greater than or equal to. So it makes to numb and. This is the current Max, so that's fine. Probably just at the start. We'll set that to be there. All right. OK. Oopsie. OK. Can't seem to move that. OK, right. OK, this is. Creating our list I presume. Just giving it some numbers to actually use, so I'm OK. I think I'm quite happy to pop that in there. Right. And then we've got the repeat here, but oh, that's possibly greyed out. That's OK, right. Just have a quick look. Set list to and it creates a list. Presumably I think that's what that syntax means. It's not totally clear to me, but I'm happy to run with it. SET Max to zero. Our initial maximum is 0. Doesn't work if it's. Negative numbers, but that's OK. For each number in the list. Yeah, numbers, even numbers odd. It's going to the current Max set. The new Max. Yeah. OK. Good. I'm going to hit submit solution and. See what happens. Ah, and passed it. Well, that's good. That makes me quite happy because I didn't. Once I built the thing. I didn't actually do anything with it. I didn't need to put it any any space, I just it was. Just there and it was. Happy to take it. What happens if I just submit a solution now? Quite and it gives you an actual Python error, and that's good. So it's starting from somewhere, presumably. Anyway, that's fine. I'll try it again. OK, first, we set a guess number in this case, set it to seven. OK, and list of input values. And cheques for this list of guesses if. Each value matches the guest number and also give feedback if the input is too low or too high and finishes and the value minus one is found in the list of inputs OK. Right. And kind of get what I'm trying to. Do here and just kind of look at the Python. Confirm that. OK. It's while Lucas interesting. To me can look at it in. JavaScript. All right. Similar. OK. Cool, right? I'm just going to run. My logic so. Setting the guest list first, that would be the first thing. In this instance, we're just setting guest number to seven, so that's fine, right? The upper list of values. If each value matches the guest number OK. This feedback is too high, right? Start constructing my while loop. Kind of annoys me that these don't move around. Because I kind of want to get out of the way. But it's OK.

Commented [RV3]: Blockly syntax: is not totally clear but first guess was actually correct

Commented [RV4]: Usability: Shows engagement and motivation.

Commented [RV5]: Feedback: Likes python error and execution based feedback.

Commented [RV6]: Blockly syntax: Looks for the code translation to understand blocks syntax.

Commented [RV7]: Clue to make pattern labels movable.

Sprecher

It's fine, sister, said the.

Tutor 1

Same. OK, so that's going to be the end thing that we do. I'm hoping that. Oh, that's why. I was being silly about what this actually means. I wasn't even noticing that it was changing last. Time. That's cool. That's fine. Yeah. I assumed that this was some representation of the programme and I feel like in the previous one it was close enough.

Viviane Rehor

That's good. Then I need to have a headline there. That actually explains it.

Commented [RV8]: Clue for adding the code module headline.

Tutor 1

That's why I was like, why is there an empty while loop? But obviously because I haven't set it up yet. That's fine. OK, I didn't actually want to do that. I just wanted to see what. Was changing so. Right. We don't actually set a guess at this point. Uh oh, we haven't set the count to either. OK, let's figure some of this right. We can set the count to 1 at the start. And we set our first guess. OK, so this is good. We've got two of these. I assume these are identical. I'm just gonna put them next to each other. Yeah. Number count. Yeah. Yeah, I think that's fine. So that establishes our first guess. That's good. If the guess is equal to the NUM to guess what we want to do then. OK, I'm going to leave that for now. We always want to do this at the end of our while loop, so I'll pop that there. To begin with.

This over here. All right. If the guess is greater than. The number to guess, we want to say that it's too high else. It might be equal to, so we don't necessarily just want to say too low. This is the place. Or if we do that beforehand, we can always assume that it won't be no, because it's not part of the do it would. Be partly else. OK. Well, I know that this can happen here. That's if we've hit it. So that's good. I am. I'm not fully convinced of this. I need to update the counter. At the end there. Yeah, not fully convinced, but let's. See what happens. OK, good. Close enough then. I didn't actually cheque what happened there report when an input number is even or odd. Hopefully that's straightforward enough. What have I got? There's my if if the mod equals zero then it's going to be even though it's odd. Uh-huh. Set the mod to be that. Set the input to be that OK. That should be what I'm looking for. That is what I would expect to see. Cool. Last one. Right given array of words which make a sentence generate and print out the comma separated words as string in reverse order. And each word reverse each word reversed as well. OK. So this looks like our setup. UM. Uh for each item letter in the list. So that's actually going to be reversing our. Words and ourselves. For each word in the list, good. What kind of definitions? We've got new word to letter. You were too. That looks like blank. OK. So that's going to be our final thing with all the words with commas. Our hearts, so that's going to put. New words at the start set new words to an empty list. OK. So I think what I want to do is this. For each item in. Analyst words. Set new words, new words, new words. Is gonna be an empty list. That's where. We're gonna put. Things in OK, so how with that? For each item word. Set our new word to be that thing to be blank. And then. For each letter in Word which is coming from here. To letter pen text. Set new word to letter.

Commented [RV9]: Experts strategy: He shows the strategy of starting with the loop, inserting ifs and then the rest. Look for his own explanation later.

Commented [RV10]: Feedback: Try it even if he is not fully convinced. Uses feedback for guidance.

Viviane Rehor

It's actually the trick. OK. Because of the puzzle pieces that were available plugging.

Tutor 1

This is going to be fun. Watch with blank string. OK, I'm gonna start grouping things by not what I can actually deal with in scope. So these have to happen inside that for loop. New words. This can happen at any level. I think this is going to be my. Possibly my final piece because we're going to construct. In new words, instead of first your new word, so something like that. Let a pen texting you. Aha, I think. This is what it should look like, so start with a blank new word. To the letter. We append text. New word that will give us a new letter and then we set the new word to be that letter. The next time we look around. Got some extra stuff on it theoretically. UM. Let's see what looks like Python. For the letter and the words. Letter is now the letter plus the new word, which will either be the previous letter or an INT string. In the first or second iteration, new Word plus letter, no new and equals letter. Yeah, so that will become. Oh, in the first instance. Yeah. Good. OK, I'm happy with that. Let me just cheque the rest of it. In this new words, insert first first words. Yep. New word rather and that will always put. The new word. At the start. Of the list. And then print the whole thing with commas. Let's see. Fingers crossed. Yes, yes. OK. You've done all of them.

Commented [RV11]: Usability: Shows engagement and motivation.

Lecturer 3

More Parsons puzzles, or do they seem slightly different?

Tutor 1

They definitely seem different I. Think just because of.

Sprecher

However easier.

Lecturer 3

Just different.

Tutor 1

How different? That's a good question. So when I think about it some more, let's see what am I doing here? That's different from past this puzzle. UM. I think. Now if I was doing a Python Parsons puzzle. Would it be any different to this really? I'm thinking about for example JavaScript, where in my passions puzzles like the options I've given. I'd have things like closing curly brackets and semicolons and things that. I'm very useful to me. I need them at some point to, for example, close off a for loop and everything. But this is all nicely done for me now.

Commented [RV12]: Comparing to PPs: No brackets here.

Lecturer 3

Some fun aspect.

Tutor 1

Just change it. Yeah, yeah, definitely in Python though it's not really an issue anyway, so I think one thing that is nice is the way that there's kind of colour coding and structure that signifies immediately when I see it. This is an if thing this is a for loop and. Once I slot those into place in the programme. I can worry about what comes next later, in a sense, and you can sort of do that with parsers puzzles, but you'll probably have to end up rearranging things because. Slotting stuff in just, well, at least in my experience, doesn't work as neatly here as well.

Commented [RV13]: Blockly syntax: Color coding is nice.

Commented [RV14]: Comparing with PPs: This has neater interaction than with swapping lines around.

Lecturer 3

OK, so even if it was. Just lines. This is a neater interface.

Tutor 1

Yeah, it means that I can construct the the kind of the outer shell programme to worry about what bits. Yeah, in a sense, because I could look at it and say, OK, I've been given 24 loops in this instance. I know that one has to run before the other, so I'll put that first one, then put the second or in another one and say, oh, I know that this if statement has to take place within this if statement or. As part of the else block, so I'll put that inside there and so on. And so it does help me plan. In a way that a pass this puzzle. I think it depends on the Parsons interface though, because I've had Parsons interfaces where you. Have 8 lines of code and eight slots, and you have to drag and put it in the correct slot. Now I don't know how many lines I'm going to use in my first for loop. So to put my second one in and I know that's tricky, but then I've also done a Parsons puzzle where you can you. It's like magnetic, so you just slide stuff in to. To make more space. I guess in my mind I'm trying to think what does this do differently to that and this feels better. It might just be because of the additional additional signalling of. This is a for loop. This is an if statement and I'll have something in here and some have something in here, whereas when it's individual lines. That clarity isn't as.

Commented [RV15]: Comparing to PPs: "This feels better"

Lecturer 3

And if if you were to consider your approach, sorry, you want me asking the. Question just ask the way.

Viviane Rehor

Ohh yeah look, go on gone. It's very interesting.

Sprecher

That you you should answer.

Lecturer 3

What's can you are you able to say? Just think carefully. Are you able to? Say what your approach was each. Time you started a new question. Like you know, I assume you. Go and read the problem definition first. Yep. And then and then what?

Tutor 1

Problem definition first. And then. I would. Go for the kind of the core function or core like programming, so something that I noticed that I was doing every time. Is that there's setup statements like data statement saying set up the list, set up the counter or whatever. Those almost always came last for me when I was putting things in place because I was focusing on this. Is the functionality this programme so for loops and while loops first and then my kind of if statements and then slotting in all the bits of that and then? Setting up the data that I need at the end, I think that in most cases that was the. Way that I approached it.

Commented [RV16]: Expert strategy: He starts with the core, for/while, if, setting up data in the end.

Lecturer 3

You you decomposed the task into a control flow plan. Yeah, that was the 1st.

Commented [RV17]: Expert strategy: Stepwise refinement.

Tutor 1

Thing. Yeah. Yeah, that's right. And in the problem specification, there isn't anything about. This is the data that we need to set up. Not usually there's like 1 qualification of for this instance. Assume that it's going to be. A number but for the most part that isn't included in the problem specification, so it passed me by a bit. Well, it was something I.

Sprecher

Right.

Tutor 1

Dealt with later and.

Lecturer 3

OK, so so you looked at the problem. Did you think about the control flow structure and then look for the pieces or did you go and look at the pieces and go ohh, I can see she wants me to use a for loop and a while loop and a.

Tutor 1

I definitely had an idea of what the flow control flow was going to be in my head.

Commented [RV18]: Problem solving strategy: He already had a control flow in his head.

Lecturer 3

Already, as soon as you looked at a problem statement.

Tutor 1

Already. Yeah, in most cases I think it just happened to match what tools I was given, but there was definitely one case where it was a little bit different to what I'd expected and I had to kind of reshuffle a bit or just had to. Adjust the plan that I had.

Viviane Rehor

For the reversal.

Tutor 1

Yeah, the reversing one, yeah.

Lecturer 3
Yeah, I thought the reverse ones a little bit unexpected somehow, yeah.

Viviane Rehor
My sister said the same and like that was the easiest I could think of using the blocks that are actually in blockly and.

Lecturer 3
Yeah, yeah, that's the problem, isn't?

Viviane Rehor
It. Yeah, yeah, my sister said this was the main point where she used the like. And you did the same the code and looked at it and see. OK, this line makes this in code.

Commented [RV19]: Blockly syntax: Especially for reversing it was a bit difficult to find blocks for my initial coding solution.

Tutor 1
Code. Yeah, yeah.

Viviane Rehor
Yeah. Then I know what it is, and then I can use it in my like. Then it matches my understanding.

Sprecher

But your sister.

Lecturer 3
Has done a fair bit of computing as well.

Viviane Rehor
Yeah. Yeah, she she is barely. Filled with her bag.

Lecturer 3
Yeah, you're finished.

Viviane Rehor
Yeah, nearly finished.

Lecturer 3
Yeah, yeah. Not the not the target audience, that's. The point. Yeah. Yeah. Umm. Hmm.

Tutor 1
Hmm, that's interesting for me. I quite liked that because it did the fact that it wasn't what I expected because I had to rethink it. I had to say, OK, this is more of a puzzle. And I've got to use the piece that. I've got to construct it. If I was the first student, I don't know.

Lecturer 3
Yeah, but first. Issue is more likely to work that way. They'll. Just go for pieces I I think.

Viviane Rehor
Yeah, yeah, because they might not even have like the. Control for in. The Microsoft.

Lecturer 3

I mean, I mean, talking to you, I'm. I'm minded to think there should be a a two step phase you give them a problem and you actually say hmm. Have a go just. Almost just typing out the control, I mean almost like, right. Write the plan that you think might might do, you know, force them to do.

Commented [RV20]: Clue to later add Step 1 and 2 because experts do it anyway.

Viviane Rehor

Yeah. Yeah, yeah, yeah, yeah.

Lecturer 3

That firstly write the plan and. Then and then have that on the side there. And then go to the problem.

Tutor 1

Because that was what I was doing. I wasn't writing anything I could hold in my head. Yeah. But that's the.

Viviane Rehor

The first question.

Tutor 1

Approach exactly, yeah.

Lecturer 3

Yeah, problem. Internal plan right now. How do these pieces fit now? Did it was there anything? Was there anything else there that helped you in any way? Anything else on the screen that was an assistance or for somebody probably at your exalted level? Maybe not, but. I mean. What's left that we haven't talked about, we've talked about the problem statement. We've talked about the pieces well.

Tutor 1

Yeah. Ohh what the the solution the testing.

Lecturer 3

No, no, no, no. It wasn't the solution. Actually, you could go back to the. Pieces pane there's. The actual programme pieces what what else is there?

Tutor 1

Are these comments?

Viviane Rehor

Yeah, you didn't.

Lecturer 3

See, I think we need to explain those with. An example or something?

Tutor 1

Yeah, because I I touched like I can't move them or anything. So. So I don't really know what to do with them. I think is my issue.

Lecturer 3
What do they look like? They might be well, comment. Yeah, I mean if. They were a comment. What would they be doing?

Tutor 1
Well, they would be. They would be attached to like the the top of a block or something to say like this one says loop through the list. So I would put that and I'll attach that to my for loop so.

Lecturer 3
You know, right. So does it look like this is attached to this and that one is attached to that and that one? 'S attached only to that.

Tutor 1
It didn't.

Commented [RV21]: Clue to have the pieces ordered underneath it. To see the connection to the pattern labels.

Viviane Rehor
Yeah. Yeah. You were never, like really using them.

Tutor 1
Ah, I assumed that they were just randomly put on the page and. I had to, OK.

Viviane Rehor
It seems like it's just like all the others are, yeah.

Tutor 1
I got you.

Lecturer 3
But but so well, so now what are you? Saying don't the yeah.

Tutor 1
Now I'm seeing that each one of these comments describes all the pieces that are underneath it. Is that right? It is, yeah, yeah.

Lecturer 3
Yes, that's what this was meant to be.

Tutor 1
See ah yeah. When I when I opened up the first thing, it looked to me like, oh, they're. Just randomly placed. On the on the page.

Viviane Rehor
Yeah. And if you don't know like the pieces or if you're not used to seeing those pieces, you probably won't get that. These are like different or.

Tutor 1
Yeah, yeah, yeah. So I'm clicking them, trying to drag them, saying, oh, that's not working. Oh, well, I'll. Just move on and sure.

Lecturer 3
I saw an even better idea. Sorry, carry on. So so now you now you've seen them. What? What do you think, Vivian? Vivian's intention.

Tutor 1
With them was probably. Probably what I can do for the most part with these is construct each bit of logic that I need. Like sort of underneath the comment and then drag it all together at the. End that make sense?

Lecturer 3
Which is the one that's the most complicated in. That sense, you know. Where you've got a green.

Tutor 1
Output. Oh, it is. Yeah, yeah. I think I. Was the one. With the but.

Lecturer 3
Yeah. So what is that green outputs all about?

Tutor 1
Ah, so OK, so it's given me. Perhaps the great outfit is like an an alternative bit that you could use here, but. Or it's not given to you as a tool, so you're. Expected to do something else.

Commented [RV22]: Clue to explain the inactive puzzle piece in the tips.

Lecturer 3
So so so.

Tutor 1
It would find the maximum. Doesn't really fit with what the comment explains of that block.

Lecturer 3
Does it say? But I mean what? What if you just had to find maximum? What would you? Need of all those pieces? If it was just a straightforward find maximum.

Tutor 1
Ohh, of course. Yeah. Yeah. So right. So I got you. Yeah. So I could use that with some other pieces to to find maximum just within.

Lecturer 3
But you think we're but but we've. Grown it out because.

Tutor 1
Why? Because you don't. Because we're doing our looping through in a different place. Yeah, we don't want to do that kind of thing here. We want to do it here. So. Shouldn't use this to do fine maximum could. But that would. Kind of overload the problem. Yeah, yeah.

Viviane Rehor
Basically the find maximum would be kind of incomplete without any loop.

Tutor 1
Yes, that's right. Yeah, yeah.

Viviane Rehor
So then the headline would be like kind of wrong.

Tutor 1
Yeah, yeah, yeah. So if I see find maximum, my assumption is OK. I'm gonna have to loop over to this, but. We're actually do that somewhere else.

Lecturer 3
So you didn't. You didn't really. Well, you either. You sort of ignored the headings cause you weren't sure what they were for and and you were a good enough programmer.

Tutor 1
Yeah, that's right. Yeah, yeah.

Lecturer 3
You you effectively just did a. Parsons puzzle.

Tutor 1
Yeah. Yeah, yeah, yeah, yeah.

Lecturer 3
Pieces together, but I mean, apart from the trickiness and.

Tutor 1
Yeah. Yeah. OK.

Lecturer 3
Any thoughts from you as we're going through this?

Viviane Rehor
So I mean, you obviously like found the solutions anyway, but like here's and it's really not like highlighted, but the tips thing does actually.

Tutor 1
And then use. The tips?

Viviane Rehor
Explain these things.

Tutor 1
It explains all this OK.

Viviane Rehor
And like how the like because you sometimes like my sisters have the same you drag the pieces. At the end and it doesn't work because they only drag if you put the.

Tutor 1
If you slip the handle yet, yes. Yeah.

Viviane Rehor
First bit or the most up? Like puzzle piece.

Lecturer 3

Yes, I I in in the. E-mail and review it. It's got a bunch of comments about. Things like tips because tip.

Viviane Rehor

Yeah, yeah.

Tip when I read that I thought oh, that's going to be something. That gives me a hint. On how how?

Viviane Rehor

Yeah, yeah.

Lecturer 3

Yeah, yeah, rather than a suggestion like.

Viviane Rehor

Yeah, not like a legend for maps.

Lecturer 3

That's like an.

Tutor 1

Yeah. Yeah, or.

Viviane Rehor

Rather than yeah, yeah.

Lecturer 3

Not how to. Use a thing that's right.

Tutor 1

Yeah, yeah, yeah. It's almost like. You would want to display the tips very prominently on the first exercise that someone did and after that.

Commented [RV23]: Clue to always show tips on entering step 3.

Viviane Rehor

Yeah, you can maybe like set a small so.

Sprecher

Just tuck it.

Tutor 1

Away. Just need to. Yeah, yeah, yeah, yeah.

Lecturer 3

Yeah. Why do you think we? Why do you think the sorry. Say again, why do you think Viviana? Put those little labels in.

Tutor 1

Probably to get me to deconstruct the problem a bit and think about each bit individually.

Lecturer 3

Each block you might. Need or each?

Tutor 1

Yeah. So like there's one here called alternate, which is like if it's even do this, it's undo this. Once I've constructed that block and that's done, I just need to slot it in somewhere else. I need to find out where it fits with the rest of the programme, which is, you know, that's good. So if I were a first year student and was just throwing all these blocks, probably there would be a bit more challenge there to to try and figure out, you know, like one thing that I did in one of the later ones was OK, scope, I know that I can't use this, this, these set of blocks outside of this for loop. Because they haven't been defined at that point. So I'm just gonna put those to the side. And you know, I can focus on one of the things. But a first year student certainly wouldn't have, well probably wouldn't have that kind of discretion. They would be able to tell just by glancing at a snippet whether they can use it here or not. So. It just. Massively increases the complexity of the problem by adding lots and lots of components in that they don't know if they need here or. Need somewhere else? Having this kind of contained bit where here we're dealing with just this one problem and all the pieces that exist can be used to to solve that one problem. I think that that that would be a helpful kind of. Deconstruction for them.

Commented [RV24]: Pattern labels: Help to deconstruct the problem, also for novices.

Commented [RV25]: Expert strategy: Using scope.

Commented [RV26]: Pattern labels: This would be helpful in deconstructing the problem.

Lecturer 3

Yeah, I'm so. Helpful in the yeah, you think helpful in the decomposition phase or helpful in the some almost building from the bottom up because we're telling them they're going to need those pieces. So let let let me put them together now or. Or is it that they're telling me I'm going to need these pieces? Let me use that to structure my programme.

Tutor 1

Well, both for me it feels more like the let me put them together now. Thing of having a a complete component that I can then plug into something else later, a less of the the kind of broader thinking about the structure of the programme.

Lecturer 3

What if? What if Vivianna had only given you the four labels and not the code first time and said? Here are some pieces. In the default position of this programme, does that help you think about the structure of?

Commented [RV27]: Clue for step 2 to make structure of the problem clear.

Tutor 1

Put them in all. That, yeah.

Lecturer 3

A programme or something like that.

Tutor 1

Probably, yeah.

Lecturer 3

If it will force you to do that first and then once you've thought about that.

Tutor 1

Yeah, yeah.

Lecturer 3
You could then press another button and. It would give you all.

Tutor 1
Block. It's quite hard to to say because I already kind of did that and would inherently do that. As someone who's got like quite a lot of experience programming of OK, I need to do this. I need to do this need to do this. I didn't then just logically cheque and see if the headings match because I don't know what I meant. That's on me, but. So for me, I'm already doing that step just as part of my solving the programme. But.

Lecturer 3
But you know how to. Do that, that's.

I know how to do it. Yeah. Exactly. And you've always gotta be thinking about the.

Lecturer 3
I mean the.

Tutor 1
Knobs programmer.

Lecturer 3
The underlying rationale here. Is that Parsons puzzles? Don't to us seem like a natural way that you decompose a programme that has lines. So so. It doesn't sort of help you practise decomposition. Yeah, because. It's sort of just not the right. It's not the right semantic structure. It's not even level exactly, it's just connectedness.

Tutor 1
Yeah, yeah.

Lecturer 3
Or is it? I don't know. I mean, I I don't think it is, but I I do know what you mean about Python having a. But you see, you still think in terms of find the maximum. And that's a bunch of different lines. You know, it's not like. One loop or and.

Tutor 1
Yeah, that's right.

Lecturer 3
The other thing I wondered is. Remember we had that long discussion when I was in Oslo about. About the functional description of a bit of code. So the it's like in the game of something, you know what? I can't believe precise details of these things, but but one of those labels might be very much related to this problem. But then the pattern you put with it. Yeah. So it might be a difference between saying find maximum. And something to do with finding the open person. Yeah. Yeah. So so the the label associated with the problem, the functional side would be find the oldest person, but then the pattern to go with it will find maps and then.

Tutor 1
Find Max. Yeah, yeah.

Lecturer 3

And so you could put the functional labels and say right? Think about how you'd structure it using these functional labels, and then when you press the next button you get the patterns as much as we know plus the. Code going with it. Actually, I can't help being irritated by this system that you can't. You can't put all the blocks, somehow show more closely that they're Co located.

Viviane Rehor

Yeah, because here you even with the information, it wasn't like immediately because this is like somehow close. Yeah, but some are somehow closer to the headlines and you don't really know.

Lecturer 3

In colour the background or.

Tutor 1

Yeah, yeah.

Viviane Rehor

Like it's not a bubble or something that you really know. OK, that's.

Tutor 1

Yeah, I was going to ask about that, but it sounds like you been around houses. Could you not just make a little coloured box that? They all see in this. Demon, but yeah. No, that's yeah. Forget, yeah.

Commented [RV28]: Clue to have pieces ordered underneath the pattern label

Lecturer 3

There is no way. Of doing this. Not necessarily completely different system or. I mean, I do feel it's a I do. Feel it's a. Bit of a. Problem that he didn't notice. And I mean I noticed cause I knew but. So I mean. You could say, well, it's not a problem because you just do training and you say that what you're fine buddies problem, you know, the first step is you're showing some some components of the problem. And you want to think about putting these in order, and then it'll be much. More obvious when. You go drunk and then this whole block of. Is with each one kind of thing and.

Viviane Rehor

Yeah, yeah.

Tutor 1

If I'd have opened the tips window, it is very clear. It's very clear about what it does.

Commented [RV29]: Clue to show the tips always on entering Step 3.

Lecturer 3

Ohh is it?

Tutor 1

But I yeah did assume that the tips window was for.

Viviane Rehor

You didn't want to, like, have. Yeah.

Tutor 1

Yeah, but if you could.

Lecturer 3
And and also because it's in the bottom. Sorry for you. Bottom right, right to the screen. That's where you don't look with your eye as well. It's it's. It's also a bit well.

Tutor 1
Just by bottom. Yeah. Just before submitting, yeah.

Lecturer 3
This is the last thing you get to. Yeah. Yeah. Whereas the one you wanted to be.

Tutor 1
Yeah, probably. Yeah. If I just open this up and you trigger the modal to start as I don't know how easy it is to tell it. It's like my first time.

Viviane Rehor
Yeah. No, that's that's. Would be easy, I guess. Perfect.

Tutor 1
And it just just pops up first thing and I can still see through the. Oh, that's a heading. Yeah. Yeah, yeah, that's good.

Viviane Rehor
What you normally have if you install in. Your app you have the tips.

Tutor 1
Right. Yeah, yeah, yeah. And that would, yeah, that would probably. I didn't really or 4 loops is fine, but yeah, you can't drag things around like that.

Viviane Rehor
And afterwards there somehow in the way.

Tutor 1
Yeah, that's right. Yeah.

Viviane Rehor
My my sister also said it's very annoying that you can't, like, put them on the side because you want. To have like your clean bit of. Code and the headlines are like all over.

Tutor 1
Could we could you extend the panel? I mean this would only work on my screen extend that.

Viviane Rehor
Yeah, actually it it is a.

Tutor 1
Out and just kind of like.

Viviane Rehor
Bit odd because on my screen it does like because I did it kind of responsive and.

Tutor 1
Take the full.

Lecturer 3
I I found this one the most annoying for some reason. Sorry I haven't got, I've not got my spectrum because you can't go that far. Across the screen or maybe if you just.

Tutor 1
Oh, I just turned it to mobile for like.

Viviane Rehor
You know, it's just like at least, but.

Tutor 1
It is, yeah, it is somewhat responsive, yeah.

Viviane Rehor
Ah, OK. At some point I think.

Lecturer 3
To the amount of dot square paper.

Tutor 1
Yeah. Yeah, so.

Lecturer 3
Yeah, because I found the square people stopped here and I didn't get nearly enough. Space, yeah.

Tutor 1
Probably cause you have an enormous resolution on your on your laptop.

Viviane Rehor
Yeah, yeah, I yours is.

Tutor 1
So I think the point, yeah, yeah.

Viviane Rehor
Also higher than mine and I didn't like because I couldn't test it on a. Bigger screen but.

Tutor 1
Yeah, if this was 1360.

Viviane Rehor
I couldn't just like.

Lecturer 3
I was only doing it on Oh yeah, I. Mean the resolution on that laptop is.

Tutor 1
Yeah, yeah, yeah, yeah. You will have well at least 1080. This is this is 4K scale to 1080 and. But if you're. On 1366, which one of my monitors at home? This is probably what it would look like. Yeah, yeah, yeah.

Viviane Rehor

But yeah, that that can be definitely solved easily, but. It's changed the resolution.

Lecturer 3

That would be good. Cause I definitely felt there wasn't enough screen estate for. That last one.

Viviane Rehor

Yeah, yeah.

Lecturer 3

Well, the one I. Did last year.

Tutor 1

And maybe an extra heading, final solution or something so there's a clear space. This is where we put things together at the end we blank.

Viviane Rehor

Yeah, yeah, it depends because some if someone would have a smaller screen.

Tutor 1

There. Yeah. Then there's after. Yeah, that's right.

Viviane Rehor

It it wouldn't work out.

Lecturer 3

It's a shame. That it's it is. A shame you haven't got more programmatic control. Over the blocks. Because in a way you'd like to have a thing that would toggle all those labels off. You know, if you wanted to. Toggle labels on and off or.

Viviane Rehor

Something you know? Yeah, I think I.

Tutor 1

Ohh that would be useful.

Viviane Rehor

Could even do this like I guess it would then like rearrange everything. So if you if you did something and then you want to have the labels off afterwards it would rearrange everything. Back to this. Where? Well, kind of and I. Could change the workspace to another workspace that has the labels or hasn't the labels. But every time you like said.

Tutor 1

But even that much space or.

Viviane Rehor

Yeah, it will reset completely.

Tutor 1

That work space, yeah.

Viviane Rehor
All the other stuff that are like, yeah.

Lecturer 3
And it's not that you want to delete them, you want to make them invisible. That's all, yeah.

Viviane Rehor
Yeah, you can definitely like.

Lecturer 3
But but if you've got more screeners real estate, it would matter, because if, if there's always a programming space. Yeah. And the component space. And just drag from one to the.

Tutor 1
Yeah, but you can just never really.

Viviane Rehor
Other. Yeah, I mean.

Tutor 1
Guarantee that you have that.

Viviane Rehor
It is possible. To have it like scrollable and then it doesn't matter if it's not a space, because then you just can't see everything but then. It's maybe also annoying.

Tutor 1
Yeah, and you'd have to, like, drag things around from places where you can and can't see them.

Lecturer 3
Can always pinch in and out can't. Or, you know, increase the. I mean if you if you could, could you programmatically allow people to zoom? Yeah.

Viviane Rehor
I said 10. Zoom. Yeah, I. Guess if you have the scrolling the zooming comes. With it. There's like a Boolean you can set. Scrollable on or off and I.

Lecturer 3
When you do the stroller, you. Can also zoom.

Tutor 1
In yeah, yeah.

Viviane Rehor
Think if they're scrolling there's also. Zooming, but I'm not sure.

Lecturer 3
Good movie. I mean. I think it's surprising how important. These things are, yeah. To get something that people just. Find horses. Great, I love.

Viviane Rehor
It. Yeah, yeah, yeah. Especially the the space and stuff.

Tutor 1

And it's like. Theoretically, really boring and nothing that you actually want to talk about, but actually actually. Practically, it's because if it's important.

Lecturer 3

Because there's not there. Then if you look at cognitive load theory, yeah, your whole load of extremist cognitive load has.

Viviane Rehor

Yeah, yeah.

Lecturer 3

Just been using a bloody. Thing, and it's just kind of irritating, you know, and the and the actual germane load of working to think about where these things should go. Is kind of not there. OK. What have you picked up so far from the conversation and the observation?

Viviane Rehor

We need to solve something where the headlines that they are somehow like more permanently explained and even like maybe. Better grouped or something? And also with the solution code and then like. I don't know if it's, I mean, maybe it's nice to see it, but like my sister also did the same thing when the pieces got like, weird or complicated, she did look at the code and see what the pieces actually get translated to and then use this in the problem solving way. And I don't know if we, I mean we could one thing I. Don't know the one, but yeah, there's some some. Times you just have to stick with what the bloody thing thing. Has to offer and like, especially for the like reversing thing. It wasn't like my sister said in the code. It's just like you have this and append that and you immediately see. OK, I have this string. I've had that. Yeah. And it's like here you have to read. It and it's different. From like from the code, but then she also came from a place where she knows. Code very well. And maybe the explanation for someone who doesn't the the explanations are the same, yeah.

Tutor 1

Because for me, definitely moving from Blockley to the programming. The reason I did that to cheque was cause the programming is very familiar to me. This is stuff that I understand and get like very, very well, whereas blocking. Yeah, well, it's not. Yeah. It just feels like this is a more comfortable, familiar environment for me to more kind of deeply understand what's happening.

Commented [RV30]: Comparing to novices: They might not need code translation as much.

Lecturer 3

Hopefully it'll show about.

Viviane Rehor

I mean, I didn't even find the remainder of things. Because I like, was so used to the modulo sign, and if I couldn't find the modular sign, I was OK. There is no modulo, but there is something, yeah.

Tutor 1

It's called remainder. Yeah, very often.

Lecturer 3

In these block. Block based languages. We find things have been turned into English.

Viviane Rehor
Yeah, yeah.

Lecturer 3
You're basically saying the kids weren't. The symbol the programme.

Viviane Rehor
Yeah, maybe maybe it. Is like. That's a problem that just appears. If you know code well.

Lecturer 3
It's a reverse expertise problem as well, you know.

Commented [RV31]: Same as above.

Tutor 1
But on the reverse expertise, I thought it would be very interesting and probably out of scope, but you know, just looking at the way an expert approaches using these versus the way. Like a complete novice does I think would be quite fascinating to. To write up. But you know. And what are? What are the common characteristics of someone who has plenty of experience in programming? What are the first things that they do and what are they using this like code example for and what assumptions that they make and so on versus that complete novice? What are they? How are they engaging? With the system.

Commented [RV32]: Variation: For a later research study

Lecturer 3
Oh yeah.

Tutor 1
It might be good to do that just with Parsons puzzles, you know.

Commented [RV33]: Same as above.

Lecturer 3
Yeah. No, I mean it. It's got me to think something else as. We've gone through this one. And that's even a simpler system than than this, in a way. And I'll just throw it out there as a as a kind of an idea it. What would happen if you simply structured the pieces of a Parsons puzzle by the patterns? Instead of justice having an arbitrary list. You you had the pieces. Of the whether you put them in order or I mean you know. I don't know. I just.

Tutor 1
Parsons group.

Lecturer 3
Yeah. I just want to.

Tutor 1
Do you have still have one solution box? Yeah, but you've got places with your parcel.

Lecturer 3
Yeah. So so you're you're encouraged to? Well, again, you're encouraged to think and you know you're encouraged. To think in terms of these pattern names. I wonder, I mean, I wonder if that's the main bit we're getting out of this, that that, a normal Parsons puzzle does not guarantee that the student things in things in terms of.

Viviane Rehor
Yeah, just breaks down the problem in smaller projects.

Tutor 1

Yes. Yeah.

Lecturer 3

Yeah. Whereas whereas what we're trying to do is scaffold that process, so.

Tutor 1

Because I was doing that even without the headings, you know, even when I ignored the headings, I was still constructing sure.

Lecturer 3

Yeah. Yeah, OK. That's what we do. That's.

Sprecher

The whole point.

Lecturer 3

That's what we do. Yeah. And. And so that's why I'm, you know, that's what we want to scaffold in the novice because that's what we do. We wanna try and train that. Behaviour. But we're. But we're not convinced that's what a novice does with the Parsons puzzle. But I suppose we haven't actually asked.

Tutor 1

I was wondering that. Because I think the work on its own is. Commendable and useful, it would be valuable to actually determine if our assumptions about how novices engage with puzzles puzzles are correct.

Commented [RV34]: Usability: Generally likes the idea and finds it useful. But has to be tested with novices.

Lecturer 3

What you could do is have a bunch of ordinary persons puzzles of reasonable complexity that involve a few pounds, so you know, complicated enough that you'd want to put together a plan of some complexity to solve it. And then ask a student. To do the same thing, just some Jack you know. Well, have a good doing. He's like done 2. Or three now. Now. Now tell me, what was your thought? Process and how did you go about? Doing that and just.

Viviane Rehor

Yeah, I think we definitely need something like this. Just plug the whole argument, yeah.

Lecturer 3

Yeah, we need something to to justify.

Tutor 1

Just to put the background in say you know. This is what students miss when they're dealing with Parsons puzzles, and this is. How we solve problems?

Lecturer 3

The next semester is going to. Be busy. I mean, I say that partly. I think partly because the. I somehow didn't find those label names hugely. Helpful to me. I think the label names are really quite subtle and tricky and. This this could. Be I don't know if this is. I mean this could be, are we just are we just recreating? Some goal labelling.

Viviane Rehor
Yeah, I mean. Partly this is where the names came from, or there was like.

Lecturer 3
Was it there some of your labels suggest? You look it off.

Viviane Rehor
For the not for the functional ones, but for the like in category words. In my sense I used those from some I think from sub collaborating, yeah.

Lecturer 3
Like ultimately will.

Viviane Rehor
Like the.

Lecturer 3
It's funny cause I I didn't. I remember Peter. Donaldson, talking about micro patterns. And he's. Yeah. Not as big as a fined maximum or. But something like you know. Input string. Yeah, you still need to know how to do that. You still need to know. It's probably going to be your main Python variable, SQL input, blah blah blah. Input an integer. You know, it's a slightly different microphone. You still need to know them.

Tutor 1
You could almost see a version of this where. You're very clear at the start about. These are the bits that are labelled. This is what this block is supposed to do, or these sets of blocks. And then a kind of further down part of it where you very deliberately don't include those, just randomly put the stuff around and the students are expected to construct their own, you know, sort things out themselves. So if we need to find a maximum, OK, so that's what it says in the specification. Instead of going and looking for the fine maximum block that's been given to me, I need to go and. Take all the pieces that I know I'm going to need to build that.

Commented [RV35]: Variant: Let the students do the grouping and come up with pattern labels.

Viviane Rehor
So let them do the group group.

Tutor 1
Yeah, I I don't know if it's really within scope, but just basically that would probably be the logical next step using exactly the same style of components, but not grouping them by any kind of sub goals and see if they well indicate that you expect them to do that themselves. If they can do that with. Block lead blocks. Then they can probably start to think about how to do it. When they're just writing code.

Lecturer 3
Blocks are in front of. Sorry, you're not, but you're saying you just pull them out of a pallet.

Tutor 1
No, no. This would be like down. The line right?

Lecturer 3
They're all in the. Pile there and you pull them into their groups.

Tutor 1

Yes. Yeah. So everything you need is here. Except they haven't been arranged by the headings and there are no headings. You. Know. So they're expected to. You know, come right. With basically what I was doing from the start.

Lecturer 3

The the the first version could be here at the headings. Yeah, press a button. All pieces are I appear organised.

Tutor 1

Basically this system.

Lecturer 3

800 number.

Viviane Rehor

OK.

Tutor 1

This would be stage one, yeah. And then the next stage would be exactly the same thing, except without the headings, yeah. I was just saying without the Hennings at all.

Lecturer 3

Yeah, I I I I thought you gonna say you got the headings you press the button they all appear and you dragged them into the headings because you know those headings are the.

Commented [RV36]: Variant: Sort pieces into given headlines.

Tutor 1

That is it.

Lecturer 3

Ones that are required but you.

Tutor 1

All right.

Lecturer 3

But you've you've that you're then being tasked to remember what should go in. You know, what does it take to do a find?

Tutor 1

No, I was just thinking of.

Lecturer 3

And then the third one could be your one where you've gotta come up with the line, but then?

Tutor 1

Yeah, I suppose, yeah.

Lecturer 3

But then your third one is pretty much what you do with the Parsons possible. I mean, they're all just.

Tutor 1

There and you've got to put them. Yeah, but it's in blocks, which I think it's just the, you know, that's just.

Lecturer 3

Do you like the blocks more than?

Tutor 1

Blocks mordon. I do? Yeah, I think so. They just conceptually feel nicer to me than just a plain earth line of code. It's just more clear about what this is and. What it does? At a glance and.

Commented [RV37]: Comparing to PPs: Blocks are more liked, now I can play with parts of lines.

Lecturer 3

Well, I think it is interesting that you can. Now play with. Parts of lines, I mean, you know, grammatical chunks. You know you can play with the expressions and things in a way that you couldn't do that before.

Tutor 1

But also things like this if statement has a, A do and an else, so I know that that is being used for well. That signals to me that there's going to be an else here. I'm going to find something that's going to belong in this piece. Which isn't obvious when you just have lines of code and there is an else which exists on its own. But like is that there is that really. Exists or that and you just.

Commented [RV38]: Comparing to PPs: If and else are already together, not a random else somewhere.

Lecturer 3

Well, that's why I'm sort of interested by the. By the. I suppose. I suppose, well, I haven't. For whatever reason, we ended up. The pieces that go with the label are sort of muddled up. But that could. Have been put into the correct structure.

Commented [RV39]: Clue to already order blocks underneath the pattern label.

Viviane Rehor

Already, yeah, yeah.

Lecturer 3

And then that's another kind of exercise. It's almost like, right? Where are my plans? Where's the pattern that goes with it? Yeah. Sorry. Where's my plan? OK, so don't I always get the return Audrey wrong here? The name is the. I mean, there's the pattern of the actual code or the whole thing, the pattern maybe, and then there's a code template or something. And then there's a pattern name or sorry. Who knows, but anyway.

Tutor 1

You think give them complete blocks.

Lecturer 3

I'm I'm sort of. I'm sort of saying give them a as complete a pattern as you can and then just say right, you need these patterns. And put this crew together. And I mean, Soloway would say that that's easy when they're one after the other.

Tutor 1

You need to fit it into.

Lecturer 3

UM. I think it's easy ish. When they're one completely inside another, but they're hardest when it's merging in the in the in the nature of that last one.

Viviane Rehor

Yeah, it would basically be because like for the final Max for example, you have the SET Max to zero and this has to like appear before you start the loop, yeah.

Lecturer 3

It's it's all like.

Viviane Rehor

Yeah. And put the other stuff. You have to like separate them if they're already like one big block of pieces, you have to separate.

Lecturer 3

This you have to. Do this. Yeah, yeah. Yeah, yeah. So. Yeah. Yeah, yeah, yeah. I mean. I mean, you can give the plan that's to find the maximum you can give the plan that's or whatever. The other piece is yeah, and. And you're gonna. Have to do that with them. Yeah, that's. Right. No, no, I agree. With that, but but you but you start. Off at least giving them. Yeah, yeah, it's all very plant based. You give them the name and the pattern and you say you're going to. Need these ones to solve this problem. Now, now get on it. Now. I don't know for these problems that almost seems too easy, but. I still think. It's probably a really useful developmental step. Because it it's focusing.

Tutor 1

Well, for one.

Lecturer 3

Right, it's this not doing the minutiae of line by line and perhaps that's what the Parsons puzzle is good at. When you're doing something as simple as a nothing more than. A fine Mac. But even to find Max is probably going to.

Tutor 1

But it's also a very clear demonstration of. This is a complete pattern and we can use this. Yeah, in this context, we'll use. It in other context. Yeah, this once it's done, we know what it does. Yeah. Yeah, yeah. And then they stop thinking about line by line of, you know, or what do I need to copy?

Lecturer 3

It it it reinforces the idea, doesn't it? And they see it.

Tutor 1

What do I need to have from the previous one? No you if you have the structure of the pattern, that's all you need. That's you need to change.

Commented [RV40]: Clue for giving them the whole pattern.

Viviane Rehor

You know, I I think so. I'm kind of like a little anxious about it being like full capacity, but it's still recording. You know, it still has.

Tutor 1

Still reporting.

Lecturer 3
I mean, I mean, there's an awful lot.

Viviane Rehor
Battery. Yeah, yeah, yeah.

Lecturer 3
Of leather here, but unless I still think there's a lot. Of useful conversations.

Viviane Rehor
Yeah. Yeah, definitely, yeah.

Lecturer 3
We think we're trying to draw it all out to pieces. With some long silences, yeah.

Viviane Rehor
I think also it does transcribe automatically. It normally does. I don't know how good it understands, but at least it's English. That's good. Well, most of the time it's Bestly supported.

Lecturer 3
And these things automatically transcribe that.

Tutor 1
What are my thoughts?

Lecturer 3
I suppose I'm left thinking that this. To me this. Has opened up many different possible kinds of puzzles.

Viviane Rehor
In different levels. Of like, yeah.

Lecturer 3
Different levels of, you know breaking down and and to sort of almost think about, well, what does each one give you and what does each one not give you. And so you know clearly ordinary line based persons puzzles don't really scaffold. They don't really scaffold problem solving. I don't think. Yeah, not in the way that. Yeah, and, but they don't scaffold it in either direction. Yeah, because because it's not organised to help you think in terms of bottom up patterns and it's not, it's not helping you decompose.

Tutor 1
That's right.

Lecturer 3
Yeah. Yeah. Whereas whereas the suggestion I had there that you gave them a problem and then the functional side names that would help the decomposition. But then if you then put the pattern names in the. Code that would almost help the bottom.

Viviane Rehor
Up. Yeah. And also this thing. Of like reusing whole. Patterns. It's like one of the.

Tutor 1
Exactly, yeah.

Viviane Rehor
Arguments that at least I had, like the argument of having reusable patterns. Yeah, like from the bottom up way kind of was some advantage of this idea that you don't have and pass this puzzles that you have like, oh, I need again this like alternate.

Lecturer 3
We're using what? Sorry. Yeah, multiple times. So so you.

Viviane Rehor
Yeah. Like kind of that they if they do multiple puzzles say ohh that's this headline again or I need again this pattern like.

Lecturer 3
So actually, actually yes, I did. We talk about this one time I remember, I mean the idea that you don't just have the primitives palette down the edge, but you have the micro patterns and yeah, and more complicated patterns. Yeah.

Tutor 1
Right. Yeah. Because in a couple of patterns appears the if it's even, then do this, otherwise do this. So doing the odd and even distinction, it's exactly the same pattern.

Lecturer 3
Yeah, I mean, yeah. As soon as you see something where you need to know that.

Tutor 1
I've gotta find out it's even wrong. Yeah. Yeah, there's one pattern you should have. Yeah, you.

Lecturer 3
Yeah, you should bother and to learn the passion. Yeah. And I just need to know that I've gotta.

Tutor 1
Should be referring to.

Lecturer 3
Do mod 2.

Tutor 1
Yeah, but if it's passes puzzles, I don't know how. How well that is scaffolding like you say and I. Have clarity how much character is that?

Lecturer 3
There's no name or but but.

Tutor 1
Yeah, your distinction that this is the pattern you. Know is.

Commented [RV41]: Comparing to PPs: Patterns are not included at all, reusability is not highlighted.

Lecturer 3
But but actually that that is intriguing because then you get to an extended point where. Yeah, you're right, the the end result of this might be that you just have a power. Pattern names. That you've learned. Abstract the the abstract names, not the functional site names, but the abstract names and.

Viviane Rehor
Yeah, yeah.

Tutor 1
Yeah, loop over list, find maximum. Yeah.

Lecturer 3
You know, so, so. I mean and and they're. The kind of patterns that I mean. We need to find somebody who speaks Hebrew and translate David Gino's book. You know I. Mean there's there's. 30 algorithmic patterns, apparently, that he he teaches and it's almost like you just have a you. And so when. You pull that out, you get.

Viviane Rehor
Isn't it just possible if it's just 30? Lines. Can't you just like? Google Translate line up able to help.

Tutor 1
I know the alphabet and of Classical alphabet.

Lecturer 3
You'll be able to. Get away with that with you.

Viviane Rehor
I mean, it's just 30 little work groups.

Lecturer 3
I wonder if his books online. I don't know. We'll just, we'll just. Buy in Hebrew. There must be somebody in.

Tutor 1
The pub who speaks Hebrew? Alice bet. Bet gimmel dalet hey.

Viviane Rehor
All we need here so.

Tutor 1
For that.

Lecturer 3
But but yeah, but just that idea and and then you just pull apart and out and it. You know the bits are plumped on the. On the paper. And then you just have to really work out. Where to put them? I mean, I love all these ideas. And and. That they they feel like good ideas. And so I'm just. Wondering for your for your work.

Viviane Rehor
You need to decide on like which ideas are.

Lecturer 3

Well, well, kind of, yeah, kind. Of kind of, yeah, I don't quite feel this one is this has. Not got the ease of. Use and it's not obvious enough quite. What you're going for yet?

Viviane Rehor

Yeah, yeah.

Lecturer 3

And but but one thing I'm seeing falling out. Is is a. Whole sequence of different kinds of exercises. Or related. Because they have this sense of giving some pieces and you've got to put them in the right place. But we're just trying to be a bit more focused on patterns and problem solving than just a puzzle that you you jump into place.

Tutor 1

And as I said, I think a lot of the usability issues that I face can be solved by just making those tips. You're forced to read them before you start, yeah.

Lecturer 3

You could see that. Yeah, OK. That would help. Yeah. The other thing I. Thought there was a tussle between. This is all an e-mail I forgot the detail exactly. The fact that this, the fact that you've decided that these need to be executable. That's good in some ways, because you can run it against test cases and it doesn't matter if the ordering is not exactly the same as the one you might have. Come up with. But the downside I thought of it was. At the moment, at any rate. You can't just have ordinary user input. Because of the way you've got the test system.

Commented [RV42]: Execution based feedback: Limits to not being able to use user input [SOLVE]

Commented [RV43R42]: Clue: Water changed.

Viviane Rehor

Yeah, yeah.

Lecturer 3

Set up you know you have to put everything into files and that that's just not. Very natural for a typical course, yeah. Am I being wrong so it's not natural for my course, because in my course we spend our time either reading from files or reading it from the user and therefore the idea of not having you know so. So if there was a way in here of actually in your test cases, specifying an input file. Which could double.

Viviane Rehor

I think I think that it was like because it's just like Python unit test and yeah.

Lecturer 3

As user input in.

Sprecher

Just just pipe.

Lecturer 3

It in I mean you know. And then and then we. Could just use info for.

Viviane Rehor

I mean it would be. Probably more difficult to set like to have another application and set up these exercises with this reading stuff.

Lecturer 3
Ohh yeah, cause you've got to actually have.

Tutor 1
A fire, but as a proof of concept.

Viviane Rehor
Because then you have more.

Lecturer 3
How do you? Sorry I've not looked at the how do you do the test cases?

Viviane Rehor
It's basically one Python file and.

Lecturer 3
It's it's only one test case cause it's it's it's always the same input data.

Viviane Rehor
Only only because it's so simple and.

Lecturer 3
That's right.

Viviane Rehor
Yeah. Yeah. And you, you already have the input data set in within the programme, so.

Lecturer 3
OK, you so.

Viviane Rehor
It doesn't need it.

Lecturer 3
You might want to be able to set three or four different input files, or as many input files as you like, and you and it has to.

Viviane Rehor
Yeah, yeah.

Lecturer 3
Run against all of those different programmes. Yeah, OK.

Viviane Rehor
Yeah, yeah.

Lecturer 3
But I think that that felt like an important thing for me as a teacher. I was.

Viviane Rehor
That you want to have input files or input even you just user input like one number thing.

Lecturer 3
A bit unnatural. It wouldn't really matter if it was user input or. Or from a file, because it's really I mean like, cause he's he's really like a file. Isn't it, you know?

Viviane Rehor
Yeah, yeah.

Lecturer 3
It just. I just read it in so.

Viviane Rehor
Yeah, like, I'm not sure like for the for the user input just input. We found that you found the blockly puzzle piece that I didn't found, but found it, but I don't know like.

Lecturer 3
Oh, yeah, yeah, yeah.

Viviane Rehor
I haven't found the other one but for files. I don't know.

Lecturer 3
If there's a well, I mean, don't worry about files. I think all the user input is so common.

Viviane Rehor
But if yeah.

Lecturer 3
Natural, say, write a programme that will take numbers until minus one and then you know something like that.

Viviane Rehor
Yeah, yeah.

Lecturer 3
I I felt. It was a bit contrived the way you. Were required to do it by creating a list with the numbers in and that's just about it.

Viviane Rehor
Yeah, but I think that with the user input definitely possible. Yeah. Unit tests like setting up it with another. Yeah. But if you if I set the unit is up, it's like definitely so and easy.

Lecturer 3
Yeah, just just for the purpose of the. Experiment to see if it works. Yeah, yeah.

Viviane Rehor
Yeah, just for the yeah.

Lecturer 3
So. So what? I'm what I'm seeing here anyway, is that you've is that you've possibly got a sequence of exercises that we've sort of discussed our way around.

Viviane Rehor
MM.

Lecturer 3
And it might be worth. I mean. You've got a recording, but I mean, you know. Jot them down. Yeah. Yeah, write.

Viviane Rehor
Write all these.

Lecturer 3
Them down, sort of in some kind of list of complexity or increasing complexity. Or something or. Or yeah, from novice to whatever you think. That looks like. Because you could imagine. Could you imagine? But the further you get away from the lines of code and the more you're working in terms of blocks and patterns, the more complicated the programmes could become. Because you're still working with individual chunks and at the. End of the day.

Commented [RV44]: Comparing to PPs: Getting away from lines lets you do more complicated Puzzles.

Viviane Rehor
Yeah. But like, I guess in the end definitely because if you get a pass puzzle with, I don't know like 30 lines or 40, it's like, yeah, yeah, you just like scrolling to the one that you need or finding like.

Lecturer 3
Make sure.

Viviane Rehor
The else to. Your if at the bottom of the thing it's like. And if it's more grouped, I'd say. Yeah, you can definitely like have like at least less screen like.

Lecturer 3
So. So you'd be you'd be. Identifying this, as it were a developmental sequence of exercises. But then deciding where you think we are in that sequence, or which one you really want to put the effort into, looking at to to get to get, get more detail and.

Viviane Rehor
And maybe if it's not too complicated. Yeah. To like let some novice try. Like I don't know how many we have of these and like let.

Lecturer 3
Or one or two of them were. No, no, we definitely gave.

Viviane Rehor
That person say yeah.

Lecturer 3
That no. Sorry, no, no, because we've sort of timed out a bit this term. It another good possible way of looking at this is to say. Early next term, so you know certainly inside January not later than that we we have ready you know we've taken a few back from drag you you could do a next iteration of this you could even just do it next week and get formula and Anna to look how it'll get one and to look at it you know and and try this process again and see what they think and see if the pattern. This is coming out more clearly. That might be a good target before Christmas, somebody else looks

at it, so I think. The next version. And then in the next two or three weeks in January. And you get something you want. Students to look at. And the attraction at that point is that they'll have fairly recently had exam results back and you know the ones who are doing so well. We'll notice something through strong practise and get better, right? So if we've created a whole raft of these and then and then we can always give them a £10. Book taken or? Something like? Yeah, we've got Amazon vouchers. Yeah. So to get 5 or 10 of them. To come in and use it, but in. A way you want to keep that five or £10 token until you've got something that. You think it's worth evaluating and you want to do a serious evaluation on it, and that might be where you have your ordinary persons puzzles and you have these ones. And you know.

Viviane Rehor
Or even like multiple of these ones.

Lecturer 3
Multiple. Oh, yes, yeah. If.

Viviane Rehor
Yeah, like, like for the different levels.

Sprecher

You done more.

Lecturer 3
Than once, yeah. Yeah. Well, if you think you can do that, yeah. I mean, yeah, I mean I. Suppose what you're saying is you know how to. Technology works now and.

Viviane Rehor
Yeah. And just like sorting them by headlines or not doing it like stuff like this, it's not hard to change or you hide them or yeah, or just show them at 1st and then the puzzle pieces come in and stuff like, yeah, that won't be that hard to do.

Lecturer 3
Or whether you can hide them or whether. You can you showed in sequence.

Sprecher

OK.

Lecturer 3
I sometimes think. Well, in some ways I do think that this technology is actually a bit limiting because you go, they have something as nice as you would.

Viviane Rehor
Yeah. Yeah, it is.

Lecturer 3
But on the other hand it's giving. You quite a lot too so.

Viviane Rehor
Yeah, yeah, it's nice with the puzzle pieces that.

Lecturer 3

So you hardly want to rewrite it. Yeah, I know. I know. Yeah. Yeah, you definitely don't.

Viviane Rehor

Spread out and yeah.

Lecturer 3

Want I mean so so so. As long as you can make it good. Enough that the extraneous mood stuff isn't getting in your way. Then it's. A good enough research. Tool, but it's probably. It probably isn't your deployment tool. If you were going to. Actually put it out there in the world and. But my God. The idea of rewriting. All of this stuff is just to relax.

Commented [RV45]: Blocky: Decision for using Blocky limits you but gives you quite a lot of free pieces.

Viviane Rehor

Yeah, because you like. The whole library that they already have in pieces is just like.

Lecturer 3

So what flexibility do you have with this stuff in the sense of what I mean, like how much flexibility does this whole Blockley system give you?

Viviane Rehor

I mean it is totally possible to write your. Own puzzle pieces.

Lecturer 3

In any shape.

Viviane Rehor

Completely in any shape with any like. I don't know like the. Holds for other puzzle puzzle. Pieces in any direction even say you can only put strings in this.

Lecturer 3

Yeah, yeah, that kind of stuff.

Sprecher

Thank you.

Viviane Rehor

It is definitely possible and.

Lecturer 3

But, but you see what? What we really want is a container for pieces just jumbled up, but it doesn't really do that, does it? Cause they they're always clicked together. They're always. Yeah. So it would be a bit misleading to somehow create your own special label piece. It was like a like a, you know, like a container, like a like a loop and plug all the pieces inside it.

Commented [RV46]: Blockly: Limitations, no container in the workspace for whole patterns.

Tutor 1

Yeah. And then they'll.

Lecturer 3

Be joined together? Yeah. No, that's the point. That's the point. Would be misleading because they actually have to have clicked, wouldn't they?

Tutor 1
Yeah, yeah.

Lecturer 3
Yeah, because I mean that would be quite a nice idea if if you created your whole big C. And it said at the top pattern. That was the keyword, and then you could put in a name for the pattern and then the pieces were just all.

Viviane Rehor
Yeah, are in there, but not feast as.

Tutor 1
Yeah, yeah.

Viviane Rehor
If they were a programme, yeah.

Lecturer 3
And and and and The thing is that the. The C shape, as it were. The container only becomes big if there's a lot of things in there, doesn't it? So you can't just make it. Big and then just place things in it.

Viviane Rehor
I mean, there is this kind of toolbox thing that you could have on the side where you can actually have categories and pieces under there.

Lecturer 3
It's not quite the same though. Is it? No, no. No, I mean.

Viviane Rehor
I think that will also.

Lecturer 3
You really are looking for a new kind of thing that that you'd have to extend the whole library with or something like that. I don't know that's possible. Is it source code you're working? I mean, it's just a library. You're working with. It's an API you're working against as well, I mean.

Viviane Rehor
Ohh I I think source code. It's it's not an API. I'm. I'm not sure. OK. Like.

Lecturer 3
No. So is there a piece of react you've loaded into your React programme?

Viviane Rehor
Yeah, it's like like a plug in for the and then you have in all the like libraries you're using, you have a folder with the. You have multiple folders with plugging stuff in them, but it makes.

Tutor 1
Sounds like source code.

Viviane Rehor

Like I mean it's. It makes the problem, the whole programme bigger to like way bigger, inserting blogging in it. Yeah. So it's I guess I can look at.

Tutor 1

You do get the code but.

Sprecher

All of it, yeah.

Lecturer 3

It's got a huge amount of stuff in there, but but it's written in what?

Viviane Rehor

Yeah, yeah.

Lecturer 3

It's JavaScript, right? OK.

Tutor 1

JavaScript. Yes, it my experience of trying to modify source code, specially JavaScript source code. It's an absolute nightmare, yeah. So it's not something I would necessarily recommend pursuing.

Viviane Rehor

Yeah, maybe I. Can find a nicer way to group some. Maybe. Maybe there's something already and I just haven't found it on Google. Someone already did it.

Lecturer 3

It's it's. Yeah, it's it's like you. Can you? You can overlap pieces already, can't you? It it if there was a way of making a peace that was like somehow.

Viviane Rehor

Ohh I did it actually in one of the exercises I kind of tried to because maybe is this reverse thing where I tried to like all the pieces that go.

Lecturer 3

But they were a bit overlapped here, yeah.

Viviane Rehor

Go and run. Overlap, but are not like like together. I try to like arrange them but they all like touch. Squished up.

Lecturer 3

Yes, I noticed they were a bit squished and it was OK, but almost a bit annoying that they were. They were. Squishy cause, but I could read them. All. So it was. OK. Actually, yeah.

Viviane Rehor

Yeah, I tried this to to see like that there are more like.

Tutor 1

I can't help myself.

Viviane Rehor
In the group.

Lecturer 3
I mean it it it can be a. Bit tricky with these block languages because. They take up quite a lot of space. The blocks are quite.

Viviane Rehor
Yeah, I mean like the list definition is like. I don't know 1/8 of the whole thing.

Tutor 1
Yeah, huge. Yeah, yeah.

Viviane Rehor
Is just the list of a nation.

Lecturer 3
That's a good. Reason to go back to reading the input. Well, once again I fall back on the fact that you get the most input when you get somebody actually. Using your system. Just helps hugely because and you. Can really have a discussion about what's going on. OK so. Your next steps are identify the different things. Different thing you decide on the one you're going to work on, you're. Going to really.

Viviane Rehor
Yeah, but for this this I mean, I could like definitely decide on one for next week where I like show. Yeah. Like one of these. But in the end, it might be valuable to try multiple decide and.

Lecturer 3
Oh yeah, yeah, that. Would be good, yeah. Yeah, I mean you could show you could just. Show. Yeah, I mean, you could just show this one again. With all the feedback that. You've got to. Get them to get or. Or you could try. Yeah, actually what? You I wonder if you could try one at least that just showed that. Functional decomposition names 1st and then. And then showed the code underneath or something like.

Viviane Rehor
Yeah. Yeah. Just like. Yeah, yeah. I don't know.

Lecturer 3
That that's. You know. I'm. I'm. I'm.

Viviane Rehor
I think it also makes it clear that they are important and that. It's like either if you use them.

Lecturer 3
So step one, read the problem. Step two, look at these.

Viviane Rehor
More like a wizard thing that you have to like. Oh, there's a problem. Next click. That's the yeah pieces. You have next thing. Now you can actually, yeah.

Lecturer 3
Know got to actually code and try. Putting it together. Yeah, yeah, that's right. Yeah.

Tutor 1
I did this. One wrong I think. When I first did it and it still. Gave me the right answer.

Viviane Rehor
I think so too, because there's only one test and there should be two because we have only the list that doesn't have the right guest. Yeah, because I only wrote one test base and.

Tutor 1
Yeah, yeah, yeah, yeah.

Viviane Rehor
It basically needs at least two test cases for you find the right solution or you have the minus one before. You found the right.

Tutor 1
Yeah, yeah, yeah.

Viviane Rehor
Solution because otherwise like that print never gets tested.

Tutor 1
Yeah, yeah, yeah, exactly, yeah.

Viviane Rehor
Or the other one.

Lecturer 3
Yeah. So that's why you need to text him.

Viviane Rehor
All my test cases. But maybe.

Tutor 1
What is easiest?

Lecturer 3
Multiple test cases because the programme has the data in it.

Viviane Rehor
I mean, it could. All right.

Lecturer 3
Yeah, well, yeah.

Viviane Rehor
Yeah, yeah, but the the input thing would definitely be helpful for that, yeah. Yeah. So I'm writing them down and deciding for one or even like an improved version of that for next week.

Tutor 1

Yeah, yeah.

Viviane Rehor

And maybe see if there's a better way to group.

Lecturer 3

If there's a way of. Putting the test input the the if there's a way of doing the. Textual input, yeah. Yeah, yeah, that would be really good. Right. You've got about two hours of transcription there.

arkinson