September 2023

Teacher Interview Procedure

PRE-RECORDING (5 mins):

- Chat with participant
- Get to know them if we haven't met

IN-RECORDING (45 mins):

Beginning Remarks:

- Purpose of study
    - Investigate experiences and perspectives of lower secondary teachers about the teaching and learning of debugging in the classroom
    - For this study, ***debugging****: Process of finding and fixing errors in a computer program,* ***errors:*** *components of a computer program that prevents it from executed as intended.*
    - Interested in debugging in a text-based language
    - Lots of questions ask about general trends; I'm aware there'll be lots of variation among your students, feel free to talk about this
    - Your expertise are valued, no right or wrong answers, just your experiences
- Reiterate consent
    - Can withdraw at any point in the interview and choose not to answer any given question
    - Interview recordings will be transcribed and anonymised
    - Once this is done, recording is deleted, and transcript is sent to you, no longer possible to withdraw your data from study
    - Quotes from interview may be included in reporting of results in thesis or academic publications
    - Will record the meeting for purpose of data collection – is that ok?

Interview questions (35-40 mins):

Warm-up questions

- What block-based and text-based programming languages do you teach your lower secondary students?
    - When do students start learning a text-based programming language?
- How many computing lessons do you lower secondary students have per week?
    - And how much of this time is spent programming?

Main content

- *First want to picture the atmosphere of your classroom in parts of the lesson where students are likely to encounter errors*
- Pre-error **(~5 mins)**
    - Give me an of what it's like in your classroom when students are most likely to encounter errors?
        - In terms of: are students working as groups or individually, on paper or using an IDE, has prior instruction been given, what are you doing as the teacher…
    - If at all, how often do students ask for your help in a typical programming lesson?
        - How often are students' requests related to struggles with debugging?
        - How much does this vary across classrooms?
        - How do you tend to feel in the atmosphere you've just described?
- *Now considering your observations of students encountering errors in their computer programs*

- Getting an error **(~10 mins)**
    - How do students commonly act or react when encountering errors in their programs?
        - What sort of emotional reactions do they display/these reactions imply?
        - If students are struggling or stuck on an error, what help do they seek (if any)?
    - What are some general techniques your students use to independently find and fix errors in their programs?
        - What are your thoughts on the effectiveness of these?
        - Are these based on your teaching?
        - Do you try to improve these over time?
- Student-teacher interaction **(~20 mins)**
- *Let's think about the point where you're in a classroom and you're going to help a student who is struggling to resolve an error in their program*
    - What are some common errors you see in students' programs?
        - Some prompts: small syntax errors in certain programming concepts, incorrect program flow with selection or iteration, type errors, incorrect print syntax, incorrect application of statements (could write these down in the Python IDLE)
        - What sort of misconceptions do you think these imply?
        - Are most errors syntactical, semantic, or logical?
        - Do they tend to be quite varied?
    - Describe the support you would provide to a student who is struggling to resolve an error in their program?
        - Can you give an example?
        - If teacher is not elaborating, show them example debugging exercise and ask how they would guide a student to help with their errors
        - Has the nature of this support changed as you've gained experience teaching computing?
        - What support tends to work well?
        - To what extent is the teaching strategies you use based on the students you're helping?
- Temporal changes **(~35 mins)**
    - Do you think students' change the way they debug over time?
        - Do you think this is due to the support you provide to them?
        - If not, why not?
        - What strategies do you try to encourage long-term?

Important research **(~40 mins)**

- What do you find challenging about teaching and supporting students with debugging?
    - How do these make you feel?
- What resources, tools, or teaching approaches would help to improve the teaching and learning of debugging for you and your students?
    - Have you seen any previous examples of any of these?

Final question

- Is there anything else you would like to add or ask before we finish the interview?

POST-RECORDING (5 mins):

- Thank for participation
- Send over bookstore voucher
- Reiterate transcript point – will get it over to them in 2 weeks