Conceptualising, Designing, and Enacting a Zone of Proximal Development for an after-school Coding Curriculum

Deepankur John Njondimackal, Lok Jie Bin
Prof Kenneth Y T Lim
NIEjr02
Introduction

- Driven by coding’s increasing uses for society (work, learning)

- Based on research of Lev Vygotsky’s Zone of Proximal Development
  - Acts as an intermediate zone of learning
  - Best range to push new concepts and skills of a learner

- Interested to know if learning efficiency in coding can improve
Objectives

● **Observable Outcome:** Teach a student coding and observe his learning process
  ○ Not possible to actually evaluate the effectiveness of a curriculum

● **Actual Outcome:** Determine the student’s Zone of Proximal Development through amount and type of help he needs on different difficulties of questions

● Modifying teaching methods depending on student’s ZPD to determine effectiveness of new teaching methods in elevating the ZPD (making curriculum more empowering)

● Document the curriculum designing and implementation experience on our side

● Narratively understand our thinking processes, the challenges we faced and the impact it made on us, particularly in our ontological understanding of coding and of curriculum design.
Methodology

- Preparation of coding lesson for one subject “Simon”
- Lasted 7½ months, usually 1.5 hrs per week
- Use of Python as main coding language
- Simon initially taught by going through and explaining concepts before trying out on computer
- Assessments (Checkpoints) to determine Simon’s ZPD at point of time to observe improvements and new challenges
Methodology

- Starting out with basic introduction of Python
- First half of lessons: learning concepts and typing programs with IDLE
- Tools used: Laptop (IDLE)
- Approximate Second half of lessons: involvement of circuitry hardware and utilisation of Python in RPi
- Tools used:
  - Laptop with a mouse
  - Breadboard
  - Resistors of varying resistance
  - Power source (through Samsung USB to micro wire)
  - LED lights
  - Push buttons
  - Jump wires
Methodology

“Checkpoints” measured:

- 16 May - First lesson, introduction to Python
- 18 June, 4 July, 11 July, 18 July, 25 July, 1 August, 8 August - Test for Simon on taught topics
- 19 September, 6 November - Assignments (Missions) for Simon to program based on learnt concepts
- 28 November, 5 December - Project, Simon to code a program to run a LED circuit with similar rules to a game “Simon”
Data Gathering

- **Narrative Inquiry:** Arriving at analyses and conclusions by reflecting on our experience and linking it to key concepts

- Descriptive reflections were written that focussed on Simon’s position of understanding, what he further learnt or had doubts about, what we covered and how we taught it, and his responses and interactions with us

- Writing the report was a key milestone of our research as it allowed us to understand long-term trends and overall development
Data Gathering

- Deepankur, as the more experienced coder, would mainly write the curriculum and teach it
- Jie Bin would act as an observer and pen down his observations
- Sometimes, worksheets were used as teaching aids by Jie Bin
- He even administered an interview to understand Simon’s profile and his perspective, as well as to understand how the curriculum could be improved
Results

Adaptation of teaching methods

● Modifications of teaching methods based on Simon’s learning style, occurs after test (Checkpoint 2)
● Before
  ○ More traditional teaching method
  ○ Concepts simply explained
  ○ Expects Simon to follow solution instructions upon wrong answer
● After:
  ○ More dialogue based teaching
  ○ Concepts more often given demonstration of application use
  ○ Simon’s mistakes pointed out and explained, Simon to modify answer correctly himself after
## Results

### Observations during sessions

- Comparison of Simon’s learning and habits between relatively older and newer sessions

<table>
<thead>
<tr>
<th>Relatively older sessions</th>
<th>Relatively newer sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent signs of inability to comprehend explanation (e.g., confused expression, &quot;huh?&quot; as reply)</td>
<td>Faster learning pace, more commonly shows signs of understanding (e.g., voice of excitement, head nodding)</td>
</tr>
<tr>
<td>Unprepared, recalls previous sessions when needed, albeit with difficulty</td>
<td>Brings phone to capture images of answers, stored to be used as templates for future tasks</td>
</tr>
<tr>
<td>More often asks questions, relatively more mistakes in program on average</td>
<td>More independent learning, finding own mistakes to resolve them, relatively less mistakes in program on average</td>
</tr>
</tbody>
</table>
Results

Observations at Checkpoints

- **Checkpoint 1**
  - Simon’s current level:
  - Simon’s ZPD: Application of print() and input() functions

- **Checkpoint 2**
  - Simon’s current level: Application of simpler functions in Python
  - Simon’s ZPD: Arrangement of line priority in programs, storage of values in variables

- **Checkpoint 3**
  - Simon’s current level: Sequence of functions arranged in each line
  - Simon’s ZPD: Circuitry building and arrangement

- **Checkpoint 4**
  - Simon’s current level: Reading Python shell functions and modules
  - Simon’s ZPD: New functions in Raspberry Pi, circuit building
Discussion & Analysis
Two things were constant in our 7.4 months...

- Refinements and changes were always being made to our teaching approach
- The basic structure of teaching Simon:

   ![Diagram]

   - **TOPIC**
     - E.g.: Operations for loop
   - **Illustrated with**
   - **Demonstrates understanding of**
   - **EXAMPLE**
     - Enter \((5 - 1) \times \left(\frac{7 + 1}{3 - 1}\right)\) into the prompt
     - Simple for loop that prints out the same text 10 times
   - **PROBLEM**
     - Create a program that takes in an integer and sums up all numbers until that integer i.e. \(1+2+3+4+\ldots+n\)
Why was it this way?

- To Deepankur a veteran coder, the Python language was more like a jumble of words that had to be quickly strung together in order to achieve a certain programming goal.
- Instead of teaching it in a series of “chapters” one after another, Deepankur found himself subconsciously comparing Simon’s current “vocabulary” at each stage with his own, and then going on one after another to improve Simon’s vocabulary by introducing the various disparate topics.
It was(n’t) all planned

- In our report, we added a curriculum outline, however in reality this was not pre-written
- Session topics had emerged on a more ad-hoc basis
- This happened mainly because a more informal relationship between teacher and student had to be maintained
- Limitations in the efficiency of Simon’s picking up of certain topics
- It allowed us to easily build a closer bond with him, due to the more conversational style of classroom
Understanding the student over time

- Not knowing his real needs we first attempted the use of PowerPoint lectures to introduce him to coding
- Seeing his boredom, we reformed by ensuring the coding environment was always in front of him
- Due to our initial view of this as a chore, we just pulled content and examples from school curricula
- Progressed discontinuously with topics that were most commonly used in coding like printing or that struck out to us more
- Instead of progressing continuously according to complexity

**Consequence:** Consistency was a big problem for our student early on
Understanding why

- Initially blamed external factors beyond our control like the frequency of sessions

- What really happened: a mismatch between our profile as a more professional coder who is often completing specific assignments and will pull large amounts of code from other sources, and Simon’s profile as a learner with no prior experience or exposure who needed to learn from the ground up

- Partial Resolution: Explaining the role of the function we taught in the greater scheme of a user using a computer
Speaking of External Conditions...

- They did play a large role in influencing developments in our intervention

- Deepankur’s unavailability on the third session led to the creation of a worksheet that Jie Bin had to use to fulfil the role of teacher himself

- Could not simply be a list or crib sheet with specific functions

- Rather a narrative in which the different topics were explained building on each other

- Had to empathise with Simon’s own profile and frame the worksheet to cater to that
Our development

- Jie Bin, through tools such as the worksheets became less hesitant or reluctant to provide guidance.
- Through constant exposure to Simon’s own learning process, particularly his mistakes and his own conversations regarding the language, stopped viewing Python as a black box and more as something he could actively control and be an authority on.
- Deepankur moved away from an overtly pragmatic view of coding in which specific knowledge was “grabbed”, quickly understood, and utilised.
- To one in which a narrative had to be crafted about the relationship between the user and the computer and the tasks to be accomplished between the both of them, and the coding concepts had to be contextualised in this larger narrative.
The Test

- When we taught him all the fundamentals, gave him a test and expected him to do it quickly and well
- Did it over 4 weeks instead and many parts only with great assistance
- Affected us and prompted us to observe his key deficiencies, such as regurgitating code instead of adapting it to the question
- This pointed out our key deficiency which was the procedural approach of rattling off demonstrations, as well as frequent monologuing
- Realized we had to step out of our comfort zone and be more emotionally invested in Simon’s learning, attitude shift
- We had to transform from pragmatic coders to instructors of an actual language
The Interview & Conversations

- Due to our relationship was of peers, was candid and freely expressed his belief that coding just a casual recreation unless you were going into the IT sector.
- However, even without structured interviews, we had numerous conversations about a wide variety of topics, from his school life to past coding programmes held in the Home and at his school.
- He compared such programmes negatively to our own after-school curriculum since those were isolated, single-day workshop/lecture style events in which the children did very little actual coding and did not retain much.
In contrast...

- We as teachers mastered a reinforcement technique in which we would continually question him when he reached a roadblock and get him to reach the mistake on his own.

- We created this approach because we were frustrated by his hard-coding and copy-pasting tendencies as well as his frequent syntax errors and were desperate to stop it.
Evolution

● Aforementioned technique was replaced with one in which we got him to explain entire programs on his own and we would explain to him what he could not

● Done in accord with the scaffolding theory in which the scaffolds had to eventually be reduced and removed to facilitate independent ability

● Especially when we were teaching him about Raspberry Pi, our roles had shifted significantly from actively instructing tutors to passively observing guides

● Exemplified through his largely independent work on the two Missions we gave him during the September and November weeks in which we were busy setting up the Raspberry Pi
The Final Projects

- We had planned for him to create a game of Simon where buttons were pressed in accordance to light flashes
- It would have marked an explicitly constructivist demonstration of his learning
- He likely could not because it involved many different tasks simultaneously
- Instead, we wrote the project for him, presented it to him and thoroughly quizzed him and got him to explain it until we were satisfied he understood how it worked
- A reversed version that was simpler to code was assigned, in which LEDs would light up based on which button was pressed up
- We deduced correctly that he would be able to independently solve such a problem because it was within his ZPD, a linear, single thread of events
Improvements

● More subjects
  ○ Learning method, prior experience and rate of learning may vary
  ○ Too few subjects for a large enough sample size to prove reliability for all cases
  ○ More applicable and useful for real-life STEM education reform
  ○ Greater challenge in customisation of the curriculum

● More than one session per week
  ○ Possible lack of exposure to coding in comparison to other events of the week
  ○ Student’s busy schedule and thick curriculum means that retention of coding topics is difficult

● Longer period of intervention
  ○ Better show a trend in the change in ZPD over time
  ○ More time to observe if there are any change in the amount of topics within the ZPD
Conclusions

- Simon’s knowledge growth is noticeable, starting from inability to use IDLE to being able to complete circuit programming at the end of the lesson.

- He continuously mastered new knowledge that was in his Zone of Proximal Development.

- Relatively more difficult topics previously undoable for him started to reach the Zone of Proximal Development, allowing for more future potential growth.
Conclusions

- We were able to build a relationship of friends with the student and have a better understanding of his social background and personality, which influenced the framework of our regular explanations and teaching approach.

- Our ability to differentiate between an ontological approach to coding and an epistemic approach to coding was shown to improve over the course of the intervention.

- We became more proficient in subtle teaching techniques that drew out more independent participation and understanding from the student.
Special Thanks

- Our school teachers Dr Maury Julien Jean Pierre and Mr Low Fook Hong
- Simon for his enthusiastic and cordial participation
- His guardians for the hospitality as well as providing a good teaching environment. Many of the staff at the Home went out of their way to assist us with logistics and to provide us with their time
- Ahmed Hazyl, Joel and any other of Prof Lim’s assistants who went out of their way through rain and shine to provide technical consultation and materials hunting.
- Our mentor-in-charge, Professor Lim Yang Teck Kenneth, whose continuous academic support and counsel, as well as transport assistance, made this project possible.
- All of you for sitting through this presentation
References

References

8. Patnaik, Esha. (2013). Reflexivity: Situating the researcher in qualitative research.. Humanities and Social Science Studies. 2. 98-106.
Image Sources

- [http://www.innovativelearning.com/educational_psychology/development/zone-proximal-development.png](http://www.innovativelearning.com/educational_psychology/development/zone-proximal-development.png)