Designing activities with digital technologies in early childhood and primary education

1 Introduction; UNESCO IITE and ECCE and me
2 About the workshop: its goals & forms
3 Digital technologies or ICTs
4 Why do we need them in school and kindergarten
5 How to use them in kindergarten
6 Key factor: teacher
7 Designing activities: why and how

I. Kalas
Comenius University
Bratislava, Slovakia
Institute of Education, London

1 Informatics education in Slovakia

• Informatics as a subject
  since the beginning: different from ICT
  mandatory since 1985; 2004; 2008 (primary)
  successes and problems
  teacher development

• supporting activity
  iBeaver international contest (Bebras, Lithuania, 2004)
1 Informatics education in Slovakia

- Informatics as a subject
  since the beginning: different from ICT mandatory since 1985; 2004; 2008 (primary) successes and problems teacher development

- supporting activity
  iBeaver international contest (Bebras, Lithuania, 2004)

<table>
<thead>
<tr>
<th>Age</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kindergarten</td>
<td>Primary</td>
<td>Lower secondary</td>
<td>Upper secondary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>little b.</td>
<td>benjamin</td>
<td>cadet</td>
<td>junior</td>
<td>senior</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

02/20
iBeaver tasks: our message to Slovak schools

our tasks cover the following components of Informatics education

- **digital literacy**
  - basic knowledge and concepts
  - computer literacy, working with applications
  - social, ethical and legal issues, security...

- **educational programming**
  - representations of solutions, processes, & behaviours
  - handling such representations
  - algorithmic thinking

- **problem solving**
  - logical reasoning
  - puzzles, riddles, problems
  - strategies for problem solving

- **data handling**
  - representations, coding, patterns, structures
  - mathematical basics of Informatics
  - data, data structures, information and data processing

---

2 Educational programming: its roles & forms

- **professional programming** and programmers
- **recently:** increased number of computer users routinely using scripting languages, models...

  **end-user programming** *(Blackwell, 2002)*
  - identifies characteristic cognitive challenges of prog.
  - suggests to study cognitive demands and operations of

- **programming is when somebody:**
  - is not directly manipulating observable things, but
  - is specifying behaviours to occur at some future time

- **why is it hard?** *(Blackwell, 2002)*
  - loss of the benefits of direct manipulations
  - use of special notation to represent future behaviours
  - use of abstraction as a tool to handle complexity
• educational programming
  = end-user programming in an educational context
  express (externalise) future behaviours and ‘play’ with them

• we want to explore resulting learning processes
  study cognitive demands of edu programming
  elementary cognitive operations
  study social and cultural contexts of edu p

• E. Ackermann, developmental psychologist
  studies how children...
  ... use and think of programming as a means for exploring
  and optimizing the interplay between a human and
  unequally responsive, surprising, or reliable devices...

• ... what programming means for children:
  
  Mastering things  tell a device what to do; make things do things
  Animating things  give autonomy; make it look out for itself
  Modulating things  do not start from scratch; use, tweak, share

---

3 What interests us... (in primary edu programming)

far-reaching goal: understand cognitive demands of primary educational programming

in this research:  - identify elementary cognitive operations in primary programming
                 - better understand what is difficult...
                 - what is developmentally appropriate
                 - how to design tasks and gradations...
                 - how to exploit the tasks in pedagogy

• A. solving problems and handling solutions
  • direct manipulations
  • handling simple behaviours
  • commands with parameter
  • repeated pattern’s of steps
  • abstractions

B. controlling an agent
  • more actors with individual behaviours
  • forever behaviours
  • interactions and events
3 What interests us... (in primary edu programming)

far-reaching goal: understand cognitive demands of primary educational programming

in this research: - identify elementary cognitive operations in primary programming
                  - better understand what is difficult...
                  - what is developmentally appropriate
                  - how to design tasks and gradations...
                  - how to exploit the tasks in pedagogy

- A. solving problems and handling solutions
  - B. controlling an agent
    - direct manipulations
    - handling simple behaviours
    - commands with parameter
    - repeated pattern’s of steps
    - abstractions
  - C. building interactive environments
    - more actors with individual behaviours
    - forever behaviours
    - interactions and events

handling simple behaviours
interpreting simple program
planning simple program
deducing the final state
deducing the initial state
filling in missing step(s)
planning with constraints
correcting incorrect solution
comparing...
describing...
explaining...
modifying...

4 Programming tasks for primary pupils

- **Necklace** (10 to 12 year olds)
  Beaver girls are dressy; they create new necklaces for every party using a special necklace machine. They only have to set a pattern, which the machine then repeats while threading the necklace. Which pattern will produce this necklace?

A  ⬢ ⬢ ⬢ ⬢ ⬢
B  ⬢ ⬢ ⬢ ⬢ ⬢
C  ⬢ ⬢ ⬢ ⬢ ⬢
D  ⬢ ⬢ ⬢ ⬢ ⬢
**Algorithmic drawing (13 to 15 year olds)**

This is an algorithm to create various drawings on the squared paper:
We choose three numbers between 1 and 9; for example 3, 1 and 6.

**Step 1.** Draw a line of the length set by the first number (that is, 3 squares on the paper) and turn right by 90°.

**Step 2.** Draw a line of the length set by the second number, then and turn right by 90°.

**Step 3.** Draw a line of the length set by the third number, then turn right by 90°.

**Step 4.** Repeat steps 1 to 3 three more times.

Which trio of numbers will create the drawing to the right?

A. 4, 1, 1
B. 1, 1, 4
C. 1, 3, 1
D. 3, 1, 3

---

**Towers (8 to 10 year olds)**

Peter and Emily got a robot to play with. It knows how to put wooden blocks one atop the other on the towers named A, B, C, D, and E. For example, if they say D, the robot adds a new cube atop the tower D.

Peter went out for a while and Emily commanded the robot to build the following towers:

When he got back, Emily told him that the last four commands she gave to the robot were D, B, C, and D. How did the towers look like before those commands?
**The Frog** *(8 to 10 year olds)*

Little frog loves to jump on the leaves of the water lilies. When she wants to jump to a neighbouring leaf, she first turns in one of eight possible directions (look at the compass with numbers) then jumps. In the figure below, you see where she started and where she finished her trip.

![Diagram of a pond with water lilies and a compass indicating directions](image)

Which sequence of directions corresponds to the red tour of the frog?

- 4, 4, 1, 0, 0, 0, 6, 6, 4, 4, 2, 2, 1
- 0, 0, 0, 6, 6, 4, 4, 2, 2, 4, 4, 1
- 5, 0, 0, 6, 6, 6, 0, 0, 2, 2, 4, 4, 4
- 0, 6, 6, 4, 4, 4, 2, 4, 1, 1, 1

---

**The Bee** *(8 to 10 year olds)*

One of the four programs below will lead the bee to the flower. Which one is it? Note that the bee cannot fly over the red barriers.

![Diagram of a grid with a bee and a flower](image)
what are the elementary cognitive operations? 1/2
what are the cognitive demands of those tasks?

future teachers perspective

our teacher students:

- estimating how difficult the tasks are...
  correct for some, wrong for the majority; contradictory
- developmentally appropriate, respecting their experience
  tasks = small stories with attractive characters
- noted that the tasks stimulate critical and analytical thinking
- were asked to list cognitive operations
  - thinking about sequences of steps
  - interpreting such sequences
  - solving problems through sequences of steps
  - working with different representations
  - backward thinking from a given state
  - planning parallel events
  - classifying objects according to certain condition
  - algorithmic thinking
our teacher students:

- Some noted that these programming tasks also... develop 21st C skills --- mostly creativity, because... many tasks have multiple valid solutions

- Identified, which elements of the tasks could be confusing to fill in missing parts of a solution, if provided with more pieces than they would need

- All noted that for solving the tasks correctly it is crucial... that pupils carefully read the assignments recommended how to refine the tasks their wording or arrangement

- Were able to explain their ways of thinking (strategies) these however were clearly ‘professional’

5 The Bee Tasks: an instrument of our study

- Imagine Logo microworld with 7 levels they represent our proposed gradation of elementary cognitive operations culminating with The Bee

1. direct control
2. interpreting given program step by step
3. building the program to follow a given path
4. reading the program and identifying the ‘result’
5. filling in 1 or 2 missing ‘bits’
6. building the program while respecting certain constraints
7. selecting correct program from 4 options

Lead the bee to the flower.
6 Data, analysis, and findings

- 128 pupils aged 8 to 10;
  1) their scores when solving 4 programming tasks
  2) the process and results when solving The Bee Tasks
  3) our observations of pupils

a) [Bar charts showing correct, skipped, incorrect percentages for different tasks]

- b1) average time in seconds spent in each level
- b2) average number of attempts and tasks per level
  what is an attempt, what is a task

[Bar charts showing number of attempts and number of tasks for different levels]
b3) how difficult the cognitive operations are
the highest portion of the incorrect decisions
in level 3. Compare it with level 6!
7 (cf. 4): more incorrect attempts,
more correct solutions
5: filling in one inner missing instruction is difficult

b4) how much did the students develop their skills
in each level – i.e. performing corresponding operations
=> by comparing how successful they were
in the first task and the last task in each level
- typical patterns of pupils’ strategies and behaviours
  <to report later>

- observing pupils when solving the tasks... and after
  4 tasks – perceived as contest, the Bee Tasks – as ‘fun’
  they liked the Bee Tasks, rating them as very easy
  according to them most difficult:
  filling in missing inner instruction
  programming with constraints – level 6
  biggest problem:
  they did not pay enough attention to texts
  one boy noticed:
  constraints lead to longer programs
  teachers were impressed by the deep preoccupation

Some final thoughts

- the topic and the methods are new and not well validated
  carefully draw conclusions!

- there are many (other) factors that determine the difficulty
  text itself; affective attitudes (of the teacher as well); ...

- we believe in ‘Informatics for everybody’ including primary
  => we need to understand the cognitive demands
  => we try to identify the elementary cogn. operations
    within... handling simple sequences of operations

- one faulty assumption (but productive)
  when two tasks are ‘similar’?

- we believe our study contributes to
  the development of better understanding
  of the cognitive demands and operations
Thank you for your attention

kalas@fmph.uniba.sk