The Kodetu dataset: A timestamped evolution of early programmers' workspaces using Blockly

Pablo Garaizar, Mariluz Guenaga, Andoni Eguíluz, Cristian Olivares
DeustoTech Learning
University of Deusto, Spain
1-slide summary
The Kodetu dataset: A timestamped evolution of early programmers' workspaces using Blockly

• Specific research question: *How is the process of acquiring basic programming skills in early programmers?*

• Data description:
  – Data source: Kodetu, a Blockly-based web app.
  – Unit(s) of analysis: user sessions and user actions within a session.
  – Total sample size: 1044 user sessions, 307608 user actions.
  – Key explanatory and outcome variables: number of «useless actions» to solve each level for every participant.
  – Example:
    • User session:
      ```
      id;session_id;lang;school;classyear;age;sex;coding_knowledge;tech_love;timestamp;actions
      266;ttgkg5;en;A;6 PRI;11;Female;false;5;2014-10-08 14:11:28;501
      ```
    • User actions:
      ```
      id;session_id;timestamp;level;outcome;workspace_xml
      11097;h7fe7v;1412770319422;1;0;<xml xmlns="http://www.w3.org/1999/xhtml"> <block type="maze_moveForward" id="4" movable="false" x="70" y="70"><next><block type="maze_moveForward" id="5"></next></block></xml>
      ```

• Current methods to analyze this data:
  • Basic statistics.
  • User actions graph analysis.
Brief explanation
Kodetu

• A modified version of Blockly (Google), a web app to learn basic programming skills:
  – Sequences.
  – Loops.
  – Alternatives (if-then-else).
Kodetu

• Modifications, levels:
  – From 10 (Blockly) to 15 (Kodetu):
    • Levels 1 to 7: sequences.
    • Levels 8 to 10: sequences and loops.
    • Levels 11 to 14: sequences, loops, and alternatives.
    • Level 15: all + abstraction.
  – Two goals:
    • Reduce the slope of the learning curve.
    • Create a 1-hour workshop for students (HourOfCode, WeekOfCode).
Kodetu

• Modifications, logging system:
  – Pre-workshop:
    • Informed consent.
    • Demographics (academic level, sex, age, etc.).
    • Previous programming knowledge (yes/no).
    • Affinity for technology (1-10).
  – During the workshop:
    • A unique and anonymous session id is created for every participant.
    • Every change in the workspace is logged.
Tell me something about you...

School name: School A
Course: 6, Primary School
Age: 11
Gender: Female
Did you know how to program?: Yes
How much do you like technology?: 6
The Kodetu dataset
Raw data

• Raw sample size: 1304 user sessions, 307767 user actions.

• Cleaning process:
  – Delete sessions marked as test-data.
  – Delete sessions with less than 5 actions.
  – Anonymise schools’ names (A, B, C, D...).
  – Standarise academic level field.

• Sample size after cleaning processes: 1044 user sessions, 307608 user actions.
Raw data

• Total sample size: 1044 user sessions, 307608 user actions.

• User sessions:
  id;session_id;lang;school;classyear;age;sex;coding_knowledge;tech_love;timestamp;actions
  266;ttgkg5;en;A;6 PRI;11;Female;false;5;2014-10-08 14:11:28;501

• User actions:
  id;session_id;timestamp;level;outcome;workspace_xml
  11097;h7fe7v;1412770319422;1;0;<xml
  xmlns="http://www.w3.org/1999/xhtml"/>
  <block type="maze_moveForward"
    id="4" movable="false" x="70" y="70"/>
  <next><block
    type="maze_moveForward" id="5"></block></next>
  </xml>
Data analysis
Course-grained analysis

• User sessions:
  – **Maximum level** achieved per participant.
  – Distribution of **time** in each level per participant.
  – **Number of actions** per participant.
  – Manual **changes between levels**
    • In Dec 10\(^{th}\) of 2014, we disabled the level menu.
    • Participants of workshops conducted before that day were able to jump from one level to the other.
Course-grained analysis

• User sessions dataset, fields after analysis:
  – **id**: numeric.
  – **session_id**: alphanumeric unique session id.
  – **lang**: en, es, eu.
  – **school**: A, B, C, D, ...
  – **academic level**: “year cycle”, eg: “5 PRI” for 5th grade of primary school.
  – **age**: numeric.
  – **sex**: Female / Male.
  – **programming_knowledge**: true/false.
  – **tech_affinity**: numeric (1–min- to 10–max-).
  – **levels_menu**: true/false
  – **timestamp**: year-month-day hour:min:sec
Course-grained analysis

• User sessions dataset, fields after analysis (cont):
  – actions: numeric, total number of actions.
  – max_level: numeric, maximum level played (not necessarily solved).
  – Per level (15 times):
    • begin_timestamp: year-month-day hour:min:sec
    • end_timestamp: year-month-day hour:min:sec
    • actions: numeric, total actions per level.
Fine-grained analysis

• Research questions:
  – How early programmers learn to program?
  – What kind of mistakes they do?
  – What kind of levels are more/less challenging for them?
  – Is the age, gender, previous knowledge, tech affinity involved?
  – Is it possible to make any classification between early programmers?
  – Does the ability to change between levels affect their learning processes?
Sequence of actions
Fine-grained analysis

• Workspace:
  – From **XML**:
    
    ```xml
    <xml xmlns="http://www.w3.org/1999/xhtml">
      <block type="maze_moveForward" id="4" movable="false" x="70" y="70"/>
    </xml>
    ```

  – To **pseudocode**:
    
    BEGIN
    FORWARD
    END
Fine-grained analysis

• Measuring “useless actions”:
  – There are many working solutions for each level.
  – Some of them use “useless”:
    • **Unused** code: actions that won’t be executed because
      the goal is reached before.
    • **Overcomplicated** code: eg, turning left three times
      instead of turning right once.
  – Users solutions can be automatically analysed via
    graph analysis.
Unused code:

BEGIN
FORWARD
FORWARD
FORWARD
TURN-RIGHT
FORWARD
FORWARD
END
– Overcomplicated code:

BEGIN
FORWARD
FORWARD
TURN-LEFT
TURN-LEFT
TURN-LEFT
FORWARD
END
First 5 levels’ solutions

• **1:** BEGIN FORWARD FORWARD FORWARD END
• **2:** BEGIN FORWARD FORWARD TURN-RIGHT FORWARD END
• **3:** BEGIN TURN-RIGHT FORWARD FORWARD END
• **4:**
  – BEGIN TURN-LEFT TURN-LEFT FORWARD FORWARD FORWARD END
  – BEGIN TURN-RIGHT TURN-RIGHT FORWARD FORWARD FORWARD END
• **5:** BEGIN FORWARD TURN-LEFT FORWARD TURN-RIGHT FORWARD END
Questions?