

What RoboTutor logs

The Apps

Although not all of them are visible, our submission comprises multiple apps:

- **FaceLogin** handles enrollment and login, and stays running in the background. It logs
 - Video clips of kids saying their names
 - Photos of kids' faces
 - All audio input except of kids saying their names
- **RoboTutor** interacts with the kid while logged in. It logs:
 - Detailed timestamped records of actions by kids and RoboTutor
- **AZ Screen Recorder** is an open-source third party app. It records:
 - Screen capture video
 - Audio
 - Screen touches
 - For the field study, we configure it not to capture camera input of kids' faces.
- **RoboFeel** uses face detection and facial analysis. It stays running and logs:
 - Face locations
 - Facial expressions
 - It does not record images of the faces themselves
- **RoboTransfer** transmits data to and from the ftp server when RoboTutor is not active. It syncs student information to update folders for all students on every tablet, so they can log in on any tablet, including replacement tablets initialized when first charged -- even when all tablets are replaced simultaneously to deploy the code drop of a new version. RoboTransfer logs:
 - Timestamped file transfers between tablet and server
 - Distance from the server (provided by the ftp protocol)
- **Android OS** takes screenshots when the power button and volume down button are pressed simultaneously. We allow this behavior but disable it when FaceLogin is visible.

RoboTutor file folders on the server

Note: The **LOCAL** and **REMOTE** folders in "Information for XPRIZE about what we log" contain examples of the files described below, but are not further organized into subfolders.

1. The **/LOCAL** folder contains information used only by RoboTutor in the field:
 - a. Folder for each student, named by student id, containing the student's:
 - i. Name narration video clip, e.g. enrollmentID3548.mp4
 - ii. Photo of face, e.g. enrollmentID3458.png
2. The **/REMOTE** folder contains data collected to send back for off-line analysis:

- a. RoboTransfer log, e.g. RoboTransfer-1342758-N34558GH.json
 - b. Folder for each tablet, named by its id, for data that can span students, e.g.
 - i. RoboFeel logs, e.g. RoboFeel-3456869-NG48C.json
 - ii. Audio .mp3 session recordings minus name narrations, with metadata markers of logged events, e.g. enrollmentID22145.mp3 (due to how FTP processes Android clients' timestamps in filenames, each user's first session recording is named just by user id; subsequent recordings have filenames that contain timestamps of when they started)
 - iii. Screenshots taken by kids, if any (prevented when FaceLogin is visible)
 - c. Folder for each student, named by student id, containing the student's
 - i. Folder per session named by session id
 1. screen capture video (without faces), e.g.:
2017_08_30_10_34_34_example_screen_capture_video.mp4
 2. json .log files (events, writing, scores, facial expressions), e.g.:
RoboTutor_release_sw.1.3.1.1_2017.09.26.17.15.33_5B02000273.json
3. The **/REMOTE** folder also contains student data both used locally and collected:
- a. Folder for each student, named by student id, containing the student's:
 - i. Json student profile
 1. Timestamp
 2. Tablet id of first enrollment
 3. Position (level and task) in each content area
 4. Eventually: Bayesian Knowledge Tracing Model of each skill

Events and other timestamped information in .json log files

- **Installation** (first use) of new version
- **Launch** tutor to exit/crash/whatever
- **Session** from login to logout/timeout/whatever
- **Activity**
 - Tutor/game
 - Menu: activity selector / debugger / activity rater
- **Tutor actions:**
 - **Type:** prompt, scaffold, feedback [possible but bad to infer from animation graphs]
 - **Audio:** speech or sound effect (stored in .mp3 with metadata linked to logged events)

- **Graphical:** screen object(s) and type (e.g. highlight, gesture) – not yet logged, but should be easy to log generically, either in a single location or in the method for each graphical action
- **Randomized choice** – to exploit embedded RCTs, we need to know the alternative(s) not selected
- **Data source** for item sequence
- **Item** – ideally standard labels for:
 - **Stimulus** (BubblePop: say/show; Akira: signs, car label, audio; Math: problem; ...)
 - **Target** = correct answer to tap, write, or say
 - **Distractors** = wrong options in multiple choice
- **Step** – most items in RoboTutor are single-step, but a few have sequences, e.g. READ pages/sentences/words, WRITE character strings, ARITHMETIC problems
- **Attempt** to perform a step (so can derive attempt #, # attempts)
- **Writing** recorded by RoboTutor's handwriting recognizer
- **Student performance:** much of the same information but in a form intended to facilitate measurement of student performance:
 - **User Identifier**
 - **Activity Identifier** (e.g. math_5)
 - **Language** (swahili vs English)
 - **Problem name** (e.g. 3+ 10 for addition; target answer for BubblePop, Writing, and Reading; 7 as the target on the car and 5 and 10 as the numbers on the lane dividers for RaceCar)
 - **Problem number** (ranging from 1 to 10 for 10-item activities)
 - **Total number of sub steps** (e.g. 32 + 55 has 2 substeps, namely 2 + 5 and 30 + 50)
 - **Sub step #**
 - **Sub step problem** (e.g. 3 + 0)
 - **Attempt #** on this problem (how many times has the student tried this problem)
 - **Expected Answer**
 - **User Response**
 - **Correctness**

- **Scaffold Used?** (Y/N)
- **Outcome** – correct, incorrect, hesitate, plus activity-specific evaluation, e.g. output of speech or handwriting recognizer
- **Unix Timestamp**
- **Prompt Type** (Visual, Auditory, etc)
- **Feedback Type** (Visual, Auditory)