Easy As Pi

Part 1

A high-level description of the progress since the previous milestone (M5)

Front-end:
General polish of the entire UI. Added error screens, an edit screen for the classroom, support for additional math problem types, statistics screen for the teacher and students, help url support. Updated the UI tests to ensure they all pass.

Back End:
Added support for multiplication, division, perimeter and area type questions. They are all generated dynamically. The algorithm that determines the next problem is complete and integrated into the route that fetches the next math problem. More fields have been added to the student progression system for additional statistic tracking. Once a student completes their daily problem set, the endpoint that retrieves the next problem stops sending problems for that student. Every day at midnight the students' daily statistics get reset to 0 and they will resume receiving math problems. Added 2 new endpoints for fetching student statistics. The first one fetches statistics for a single student, the other fetches statistics for all students in a classroom. The algorithm that generates incorrect solutions has been iterated on and has made the multiple-choice answers trickier to guess at.

Major decisions and changes in the scope of the project (M0)

Instead of generating and storing problems in the database, then fetching from there, the algorithm now generates a math problem on the spot on the server-side.

The contributions of individual team members to the work done so far

Quentin Added statistics screens to the client app, polished the homescreen for students and teacher, added support for multiplication, division, perimeter and area questions on the client. Updated UI tests. Added help url support. Added error screen support. Added the possibility to modify classroom information.

Arjun Added functionality for multiplication, division, perimeter and area questions. Added additional fields to student progression for better statistic tracking. Added 2 routes to retrieve student statistics. Integrated the next problem algorithm into the route that fetches the next problem. Further developed the algorithm that generates incorrect solutions.

Noor Added multiple features to classroom service, classroom and mastery model (i.e. a field with the number of daily problems set, service to update a classroom field by field, get a classroom, get stats for a classroom etc.) and also added routes for those services. Improved

the next problem algorithm to fit into the other services. Added trigger functions to MongoDB cloud to reset students' daily progress counters at every midnight.

Arman No contribution