Learning Equality -- KA Lite: Khan Academy, Offline

The Foundation for Learning Equality (FLE) is a non-profit organization that is committed to enabling every person in the world to achieve a quality basic education, through supporting the creation and distribution of open educational resources, and facilitating their use inside and outside of classrooms around the world (https://learningequality.org/), with special emphasis on reaching disconnected and disadvantaged communities.

Since 2012, FLE has been developing and supporting the deployment of KA Lite (https://learningequality.org/ka-lite/), an open-source, cross-platform learning platform that provides access to Khan Academy educational materials -- including videos, exercises, and coach reporting tools -- to users in offline settings, on low-cost hardware (such as Raspberry Pi's and Android tablets). KA Lite has been installed in over 140 countries around the world (https://learningequality.org/ka-lite/map/), and is being used by hundreds of US-based and international organizations as part of their educational initiatives. More recently, FLE has been working with its partners to extend KA Lite into a more general offline content distribution platform, and adapting it to serve the needs of diverse learners in countries with limited ICT infrastructure.

Getting Started

To get KA Lite set up for development, follow the instructions in the wiki:
https://github.com/learningequality/ka-lite/wiki/Getting-started
Note: You should be working from the develop branch, as that is the code under active feature development.

Tips for a successful GSoC application:

- The best way to contact us with questions is via info@learningequality.org or in our HipChat room at https://www.hipchat.com/gzQfGFgv1 (since then others can answer your questions too, in case the listed mentor is not available). Please do ***not*** email us with GSoC inquiries at dev@learningequality.org, since that has wide distribution.
- When contacting us, please have *specific* questions (not just "any pointers?"), having already attempted to get started yourself.
- In the application, explain your relevant experience, and how you would approach the project you're proposing.
- Whenever possible, submitting code is a valuable way to demonstrate your ability and interest. This could mean directly opening PRs for Kolibri or KA Lite, or sending us links to code uploaded elsewhere (if it's not something directly part of Kolibri or KA Lite).

Google Summer of Code Project Ideas

The ideas listed below are examples of projects you could work on with us, but we're also very open to considering your own ideas and proposals for KA Lite, or to chat more with you to see how your interests and skills could fit in with our roadmap and project needs. Just reach out to us at info@learningequality.org if you have an idea and want to run it past us for feedback!

- Idea #0: Propose your own improvements/features for KA Lite
- Idea #1: Recode Khan Academy videos into a low-bandwidth vector format
- Idea #2: Improve accessibility of the KA Lite platform for screen readers etc
- Idea #3: Performance optimization of KA Lite for low-powered servers
- Idea #4: Peer-to-peer distributed database replication engine
- Idea #5: Integrate Khan Academy Computer Science platform into KA Lite
- Idea #6: Add an SSH-based "Remote Assistance" feature to KA Lite

Idea #0: Propose your own improvements/features for KA Lite

Useful skills: Some combination of Python, Django, Handlebars, JS, Backbone.js, jQuery.

You can view outstanding issues here: https://github.com/learningequality/ka-lite/issues
And some recent projects here: https://trello.com/b/3ZXutkYn/fle-projects-winter-2015

After you've played around with KA Lite (reminder: checkout the *develop* branch, to see the latest), if you see something you would really like to add, put together a concrete proposal and run it past us.

Idea #1: Recode Khan Academy videos into a low-bandwidth vector format

Useful skills: OpenCV, Python, video processing

Potential mentor: "Richard Tibbles" <richard+fle@learningequality.org>

Difficulty: Medium

Khan-style videos consist of a sequence of various types of operations: pen strokes, erasures, image overlays, and window panning. Replacing the higher-bandwidth video with a more compact representation that can be replayed in the browser can drastically reduce both the bandwidth and the the storage space required, and thereby reduce cost and increase access. For information on some relevant related work we could hopefully build off of, see "NoteVideo": http://www.youtube.com/watch?v=_LT46tOOipU / https://dl.dropboxusercontent.com/u/1080765/notevideo.pdf

The main ideas are to:

1) Devise a way to encode Khan Academy style 'blackboard' videos to be just a background colour, and then encode the build of vector paths in a timestamped way + audio. One possibility for this is to use a library like fabric.js and use the HTML5 canvas to display the videos, then record line drawing events for playback.
2) This part is more challenging - it would involve using computer vision (python-opencv) to take existing Khan Academy videos and encode their video content into this new vector based format (which should be far lower bandwidth than even a well encoded video).

Recommendations for preparing an application:

- Required: Describe how you would attack this problem, and what tools you would use.
- Recommended: Submit samples of code you've written using python-opencv.
- Optional: Write some sort of basic processor for a KA video, to (for example) track the mouse pointer location.

Idea #2: Improve accessibility of the KA Lite platform for screen readers etc

Useful skills: HTML, CSS, familiarity with Accessibility standards.

Potential mentor: "Richard Tibbles" <richard+fle@learningequality.org>

Difficulty: Easy to Medium

One of our core values (https://learningequality.org/about/values/) is "Design for Everyone", and we aim to make our work available to as many people as possible, which means making it accessible. We still have a long way to go on this front, starting with a platform evaluation and list of recommendations, and then implementing these recommendations.

A good proposal would include biweekly lists of portions of the KA Lite app that would be made more accessible, and proposals on how to do that. The initial part of the proposal would outline the general strategy, which could then serve as the start of a guide document for future developers as they implement features to maintain accessibility in the codebase going forwards.

The idea would be to be making these changes iteratively, so that Pull Requests would be being made regularly, being reviewed, and being merged in, so that you are always working on the active development branch. This will help avoid code divergence and messy merge conflicts.

Recommendations for preparing an application:

- Required: Describe your experience with accessibility standards and web development, and how you would approach improving KA Lite's accessibility.
- Required: Open 3 accessibility-related issues for KA Lite on GitHub.
- Optional: Submit a couple of pull requests with accessibility improvements to KA Lite.

Idea #3: Performance optimization of KA Lite for low-powered servers

Useful skills: Python, database optimization (SQLite, especially), Django.

Potential mentor: Aron Asor <aron+fle@learningequality.org>

Difficulty: Medium to Hard

KA Lite was originally designed to run on very low-power hardware, like the Raspberry Pi, and has stuck close to this principle as the project evolved. However, as we continue to add features to the platform, it has started to slow down for lower-end hardware being used as servers to provide access to many simultaneous client devices.

Some of the performance bottlenecks include:

- The SQLite database backend being used (for cross-platform portability), especially as we start tracking more and more data and deployments get larger.
- Content and topic tree metadata, which is loaded into in-memory caches at startup, as more and more content is being included.
- Overhead added by the Django machinery we're using; finding ways to streamline the request processing pipeline could help.

Recommendations for preparing an application:

- Required: Run some benchmarking/profiling tools on KA Lite and make recommendations for where optimization efforts could be focused, and outline the primary steps you would follow.
- Optional: Submit a pull request with a performance improvement to KA Lite.

Idea #4: Peer-to-peer distributed database replication engine

Useful skills: Python, Django, sqlite, algorithms, CouchDB, distributed systems

Potential mentor: Jamie Alexandre <jamie+fle@learningequality.org>

Difficulty: Hard

The typical use case for KA Lite is operating offline, which means the learner data being recorded is stored in a local offline database. As many users want to be able to view aggregate statistics and learner analytics in a central online repository, we built a mechanism into KA Lite for synchronizing the local offline databases with our central cloud server whenever there is connectivity. This system has been hugely powerful, but also has many failings that we wish to address. This project is to redesign the data schema and syncing architecture that will underlie future versions of KA Lite -- making it more robust, more flexible, and more efficient.

More details:

- https://trello.com/c/SaLp4gUO/10-database-flexmodels-sync
- https://docs.google.com/a/learningequality.org/document/d/1YiiOBimlh4-6zjXHreFp-gt7Cw3bv-Qey2uLSJoHwXM/edit

Recommendations for preparing an application:

- Required: Review the documents/threads linked above, and in your application provide comments/questions/ideas based on the existing brainstorming.
- Required: Describe how you would approach the problem, and any experience you have that could help inform the project.
- Optional: Prototype/design some component of the system, explaining performance tradeoffs/considerations and how it helps address the design goals described in the document above.

Idea #5: Integrate Khan Academy Computer Science platform into KA Lite

Useful skills: HTML, Javascript, Backbone.js, Django.

Potential mentor: Richard Tibbles <richard+fle@learningequality.org>

Difficulty: Hard

KA Lite currently includes Khan Academy's videos and exercises. They also have an awesome interactive JS-scratchpad-based CS curriculum that would be fantastic to integrate!
https://www.khanacademy.org/computing/computer-programming

The code for Khan Academy’s JS coding environment lives here:
https://github.com/Khan/live-editor

A first pass integration has been completed, with the code editor loading on a standalone page here: https://github.com/rtibbles/ka-lite/tree/COMPUTERS

The final integration would have to wrap the Computer Science platform in a Backbone view so that it can be integrated into content navigation alongside our other content.

To propose how you might approach this, you will need to consider:

- How this code could be integrated into the KA Lite codebase. Where would it live? What might the templates look like? What would you need to change?
- How would you integrate the various components of the KA CS tutorials -- videos, interactive code examples, narrated audio tracks -- into KA Lite, for offline use?

- How to get the data needed from Khan Academy -- what API endpoints do you need to work with to get the tutorial code, the tutorial page metadata, the videos, etc? Where/how would this data be stored/accessed in KA Lite, efficiently?
- If a user modifies the code, or creates their own JS program, using the framework, where would their data be stored? What would be needed for that?

Idea #6: Add an SSH-based “Remote Assistance” feature to KA Lite

Useful skills: Python, networking, possibly HTML/Backbone.js for frontend.

Potential mentor: Jamie Alexandre <jamie+fle@learningequality.org>

Difficulty: Medium

In many deployments, people want us (or a project admin) to connect in via SSH and diagnose issues or help set something up. Since this is sometimes via a 3G dongle or behind a firewall, it's often not possible for us to directly SSH into the server.

We have set up a very basic "reverse SSH tunnel" system (see e.g. https://www.howtoforge.com/reverse-ssh-tunneling) that allows a local user to initiate a connection from their server to our central online server, and open up a port back to itself over this connection. We can then SSH into the central server and SSH from there through the open port into the user's server to provide support.

Our current approach sucks, for several reasons:

- It relies on SSH being installed on the user's server, whereas with Paramiko (http://www.paramiko.org/) we should be able to do this in Pure Python.
- It picks a random port from the central server to use for mapping, which could lead to collisions. It could instead first request an open port and use that one.
- This should all happen from within the user interface, on both the local and central server: User clicks to request help, notification pops up on central, where an admin clicks to open an SSH terminal back into the server to provide support. The central server SSH interface could even be via a web-browser-based terminal interface, for ultimate seamlessness.

More details at: https://trello.com/c/D8O2IFd8/21-ssh-tunneling

Recommendations for preparing an application:

- Note: There are already a couple of people working on implementations of this, so you may want to consider other projects instead to have a better chance.
- Required: Describe how you would approach the problem, and any experience you have that could help inform the project.

- Optional: Prototype any parts of the system you’re inspired to work on, and include a link to the code.