Moai Sound API Product Requirements

We'd like a simple, cross platform library for playing sounds and music that we can add to our open source middleware project.

Instead of asking you to handle integration with the Moai codebase and Lua wrappers, we'll treat the library as a 3rd party contribution and write the integration layer ourselves. This means the library will be usable as a stand-alone audio solution as well.

This library can be written in C or C++ and link to code written in Objective C or Java (as necessary). We'd like as simple a build model as possible: it should be easy to drop into our IDE projects (xCode, Eclipse and Visual Studio). It should also be easy to roll into our master makefile for the whole project.

We are contemplating three stages of development to meet our immediate, near and long term requirements.

1. Immediate requirements:
    a. Play an uncompressed sound from a file or from a memory buffer (for sound effects).
    b. Play a compressed, streaming sound from a file or a memory buffer (for voice tracks and background music).
    c. Channel and channel mixing model for playing multiple sounds at once to any number of devices.
    d. Solid seek and loop playback controls.
2. Near term requirements:
    a. MIDI player.
    b. Support for SoundFont based sound banks.
3. Long term requirements:
    a. Support for additional DSP effects (echo, reverb, etc.).
    b. Support for MIDI events requiring software synth.
4. Longer term requiements:
    a. 3D positional sound and effects (Doppler).
    b. Support for network streaming (streaming audio, voice chat).

Target platforms are (short term):
1. iPhone
2. Android
3. Windows
4. OSX
5. Linux

Clearly, the devil's in the details. My background isn't sound, do I still don't know what I don't know. FMOD has a pretty impressive feature set and an architecture designed for configurability and cross platform. I'm pretty impressed with what they've built.

I’d like to move forward with an architecture that anticipates a full set of features, like FMOD’s. That said, I’d like the architecture to be as loosely coupled and modular as possible. Since we’re exposing the functionality through Lua, there are a lot of things the end user will be able to do themselves. For example, it’s trivial to write a resource cache in Lua, so we don’t need one in the library. It would be fine (and preferable, actually) just to deal directly with individual instances and memory buffers.

For the purpose of this project, let’s estimate each stage separately. As we move down the road of cloning more and more of FMOD’s feature set it may make sense to start charging a license fee. I can think of a number of ways to structure that; let’s cross that bridge when we come to it.

In terms of project process, send me an estimate. When we get going, I’d love to see an architecture proposal with a straw man version of the key pieces of the API. I’ll defer to your experience, but there may be some decisions we can make there that will ease the integration into Moai and Lua as well as some ways we can leverage Lua for things like resource management.