1 GsoC application for Mixxx

Hello, my name is Anastasis Grammenos. I am a 22 years old student at Csd AuTh, in my 3d year. (Actually is the 4th year but due to personal reasons I was not enrolled in classes for 8 months, so I missed 2 semesters).

I grew up and live in Thessaloniki/Greece.

1.1 Musical background

I play the guitar and I know my way around a piano. Also in the past 2 years I've been jamming with electronic music and I am really into analog gear. As far as DJing goes, I don't really know how to do it, since usually when I pick the music it sets people off! But I still like doing in for my self from time to time.

1.2 Programming background

I've done work in C, C++, python and a bit of Java (although I hate it), mainly for school projects and my own satisfaction. I really like the "low level" approach C/C++ offers and I think it's a great language to base a project on.

I know my way around a database, since I work in a photography shop, where we have one made by a coworker and from time to time we need to run strange queries and fix some mistakes other users might do. Currently we are working on a replacement of the old DB with a new one with PostgreSQL, witch is way more advanced that the one we have right now.

I am a full time Arch Linux user in both my laptop and desktop, with the exception of photoshop and other adobe programs I use for photography reasons.

That said I do not consider my self a linux power user, I just try to become one. (there are still a lot of manpages to read)

I have moderate experience with git, but since I never used it in a big project I don't know all the rebase stuff, reverting changes etc.

1.3 Mixxx specific requirements

1.3.1 Scons

I downloaded scons, and i've been playing with it, building my old C code. Since it's basically a python script, I don't think it will be hard to get to know it.

Also I looked at your SConstruct. WOW, that's cool!!! I didn't even think there was such an amazing tool a week ago.

1.3.2 Git

As mentioned above I have knowledge of git but only the basic stuff. Sadly I deleted some old repos with C and C++ but here is my github if you'd like to have a look at it. (I use it mainly to keep track of my config files) `https://github.com/gramanas`

1.3.3 Qt

I've seen some tutorials about it, but I can't say I know it. I am of course willing to learn everything since I believe it's a great tool to have in my toolbox. Since it's based on C++ I don't think it will take a lot of time to get acquainted with it.

1.3.4 Portaudio

Well, I know nothing about it! It has something to do with audio i think :) I will look at it tho in the coming weeks.

1.4 My ideas about Mixxx

The following are things I am interested to tackle for the summer

1.4.1 Link controls

Basically this feature request: link controls (control two knobs with one mouse)

1.4.2 Crates

This is the feature you proposed on the wiki. I love the idea of tagging stuff, and I believe crates can make mixxx stand out if implemented correctly.

As I see it, there needs to be a database of user defined tags = crates. This part already exists.

What **NEEDS** to happen is smart crates: automatically add music depending on user defined stuff in a selected crate when scanning the library for changes.

After that mixxx needs a search bar for crates with basic regexp functionality (E.g. (hip-hop AND metal) OR (instrumental AND metal) would list songs that are both in hip-hop and metal crates or in instrumental and metal crates basically, you'd get a playlist of hip-hop and instrumental metal)

This can make or break the music organising before a gig for many DJs.

As a bonus point I'd like to add what I call "tag tree" witch depending on the crates each song is in would be like this:

- Song.mp3
  - Tag 1
    - Tag 1.1
    - Tag 1.2
  - Tag 2
    - Tag 2.1
      - Tag 2.1.1

So let's say we have a song.mp3 in crates metal and instrumental. Then we can create a subcrate of death metal nested in the metal one.

The idea is that you have crates for the very basic stuff e.g. Metal, or 80's and the subcrates for more detailed tagging.

Remember a song can be in as many crates as the user wants.

A new UI just for this could also be implemented later on.

1.4.3 Step sequencer

This is an idea I had for quite a while, and I think I got it when I was watching a dude perform on traktor in YT.

The design is based on Doepfer Dark Time, witch sadly I don't own (yet).

The idea here is that the user adds Cue points and then he can make a loop based on them using this step sequencer.

I think of it as a popup window (like VST's in DAW's) where the user can choose what cue point will be triggered for each of the 16 steps available. He can skip steps or mute playing for some of them. This makes a really in depth beat making thingie which one can have loaded in one of the 4 decks with some premade cue points and sequences (save and load setup must be implemented for this to work good) to accompany other songs.

I think this is by far the hardest one to implement and I am not sure I can do it alone in 3 months. But I list it here cause I might inspire someone else, who knows.

If you are interested for this feature I can and will try to make it to master by September but as I said, help will be needed.

1.5 General notes

I've never been a part of something as big as mixxx. I bet it will be an amazing experience. As far as I can tell, I would like to be part of the dev team, since I find it a really nice piece of software and a great way to start my ”career” as a programmer. (I don't like the word career, cause it makes people step on other people for their benefit regardless of the pain caused in others, this -among other- is why I am interested in the free software community.)

I believe that, just like music, coding can be and is a form of art. I like code that follows a standard (spaces or tabs, not both, is one example of what I mean by code standards). I also like code that has comments because it's accessible. Skimming through mixxx's git repo I saw that the code is not very well commented. As a side quest for the summer I'd like to organise it (since it will immensely help with understanding how the program works, as a whole) by adding / _comments and /* explanations_ * before classes and functions, to make the mixxx development an easy task for the years to come.

For this task I would require some guidance from the veterans of the program. I think a good time to do this would be in the month I have to get acquainted with the code and the dev team before the summer starts.

1.6 In case anyone is interested

I use Emacs as an editor. I also use it as an IDE for C/C++/python etc.

I am familiar with the terminal and I try to do everything in it. For example as I write this in Emacs's org-mode I have tmux open with cmus, mutt and weechat for music, mails and IRC respectively.

Also for the Windows side of things I think it should be noted that I maintain about 15 windows PC's (thankfully I'm not alone) in the photography shop that I work so I am familiar with some of the common problems a program might face when run on windows.

2 Final words

Anyway, I'd like to hear your thoughts on the aforementioned, especially my ideas for gsoc. Any and all questions are welcomed.