1. Introduction

Currently, most music production software is a native application, that being, that it does not run on or even require the Internet. The benefits of this is that it allows to be extremely powerful, allowing the software to make full advantage of the computer's capabilities. They often have in-built synthesisers, drum machines and effects units; along with the ability of being able to plug in external machines it allows the client to produce and create any imaginable genre of music. However, even though modern applications strive to make the learning curve as minimal as possible, the use of them to a beginner can be quite daunting. This and the fact that it is near impossible to be able to collaborate on a single music project in real-time unless in the same room give rise to a few, but very important, disadvantages.

The rise of the Internet saw a slow and steady increase in web-based applications. Only recently with the advancements and wide acceptance of JavaScript and its in-built capabilities of creating manipulating sound has it really began to pick up. There are a few powerful and in-depth web-based music production tools and a whole host of simple, standalone, web-based synthesisers, drum machines and effect units. However, very few, if not none, make full advantage of the raw capabilities of the internet, communication. I believe there is room for a web-based music production tool that allows for real-time collaboration. The only requirements a user needs is access to the Internet, minimal knowledge of how plug in beats of a step sequencer and manipulate them (although should be intuitive enough to pick up without prior knowledge) and someone to create music with!

POTENTIALLY CHANGE TO BACKGROUND/MOTIVATION

2. Business case

2.1 Background and Motivation

Music production software has taken many forms since the birth of computing (REVISE). They allow clients to write, produce and create all different genres of music, only needing a computer and the software. Modern adaptations strive to make it as easy as possible, with little-to-none prior knowledge of music needed to get going and a wealth of in-built synthesisers and drum machines, often replicating real world tools to a high standard. However, the professional software is very costly and requires a relatively powerful computer to operate effectively on. They also are very solo artist orientated. That meaning, unless in the same room as the producer, it would be near impossible collaborate on the project. One would imagine, that the spawn of the internet would have meant a rise to many web-based applications to create music, as it did with many other industry software. However, only recently with the wider acceptance of JavaScript, with the computing power to effectively create and manipulate sound in-built, has web-based music creation applications have risen.

The advantages of having web-based music creation software over bespoke native applications is the same as the advantages of the Internet in general. Anyone with access to the internet can create music. Whether it is communication of computational power, the internet allows collaboration on a world wide scale. The amount of web-based production software, synthesisers, drum machines has steadily been increasing. Most however, do not harness the true power of the internet, communication. I see that there is a real opportunity to create some music production software that allows for real-time collaboration. This meaning that not only is the access to the software is only restricted by the user having access to the internet, it also allows for two people, from anywhere in the world, to collaborate on the same music project at the same time.

2.2 Objectives

- Two clients can simultaneously click and create music together using the step sequencer that it is available by access of a web address.
- Remove the need to send music projects back and forth or to have to meet up in 'real-life' to collaborate on a music project together.
- The user-interface is intuitive enough to make it possible for any skill level to compose a melody/beat.
- In doing so users are able to 'sketch' out a beat effectively and efficiently for later consideration
- Can save and download the project as a audio file.

  CONTINUE..

3. Project objectives

1. Web-based application that has a step based sequencer with pre-loaded samples.
2. Within the application two users can simultaneously click to create a drum pattern and melody.
3. Users can then download their musical composition as an audio file.
4. The user-interface will take the form of the time signature split into 16 equal “steps” for each of the samples. Each step is equivalent to one note within the bar. By turning these steps on and off it will in turn make the

sample play or not when the step is initiated as bar plays. The users can use this build up patterns of beats to create songs.

4. Initial scope

1. I will use on-line project tracking tool 'Trello' to plan my project. In this I have boards that have general things that are needed for the project and within further columns in which I can move cards, which are more specific tasks to the board, from left to right to track progress. I will also do this to track the work my fortnightly sprints. I will use boards in the following ways:
    (a) To have board on the project initiation which is what I need to do to get the project at a level where I can start working on the code. This include cards with check-lists for setting up server, bare bones code base, project documents etc.

    (b) To have a project backlog board which houses all my project ideas, requirement gathering for them, when requirement gathering is done moves to ready for story point estimation and then finally sprint candidates.

    (c) To have a board that tracks my current sprint. This will have columns for Sprint backlog (list of user stories for the sprint), In progress, a column for QA-ing (making sure it hits requirements and further testing of functionality), ready for release and done (a card will move into done when it is deployed). I have an extension to Trello which allows me to assign and track user story points for each user story. This will allow me to refine my assigning which ultimately will give me a better idea for how much work I have and how long it will take. I plan to show progress in a burn down chart.

2. A basic step sequencer in which user's can program simple beats and melodies in to create patterns and songs

3. The user's should see each other's actions on the screen as they happen. This will give the 'real-time' feel as they can then respond to the other user's action accordingly.

4. I hope to be able make it possible so that the user's can choose from a drop down of samples for each sequence so that they have complete control of what sounds they can use to create the song.

5. The user should be able to add and remove sequences from the whole sequencer, so that they can include or get not include as many instruments as they want.

more detail about project objectives - provide further information about the objects in a more informal manner scope can include what you'll do and what you'll produce

5. Resources and dependencies

The project isn't currently dependable on any external resources or products. I have provided the server in which it is run which is funded by myself. Potentially, for some desirable features, I might use a pre-existing JavaScript library for some desirable features to make development faster. However, I intend to avoid the use of them at all cost throughout my project and to create everything myself.

I may use the front-end framework 'Express' that is heavily used throughout many NodeJS projects, however this is isn't decided for certain yet and even if so, the project is very stable and with a large following so it wouldn't cause any problems.

More information on the Express framework can be found here: http://expressjs.com/

6. Method of approach

My software development process will follow that of an AGILE approach using SCRUM methods. That is, to release working minimal viable products early and often.

The first stage of iterations will be to get the core product working and usable. I want a step-sequencer that a user can create simply pattern and melody using basic samples.

The second stage of iterations will be to get the operational transformation algorithm in place which allows for the 'real-time collaboration' portion of the program to work.

Once these are in place, that is the core functionality of my web-application done and I would like to focus on incrementally improving the product and adding desirable features.

Possible technologies are NodeJS and within the JavaScript Web Audio API, Express framework. Also HTML and CSS for all front end styling and layout. I haven't decided whether I will be using a relational database or a NoSQL database yet for storage of samples and potential storage for user's session/work. I need to revise and my data that will be stored and see which suits best. I'm leaning towards a NoSQL approach due to the fact the data will be of different types and if I am storing user sessions the data will be unstructured or at least difficult to restrain into a table. If I do decide to go with a relational database I will use MySQL as I've had previous experience with it, otherwise I will use a document store NoSQL database such as MongoDB.